Descent Graphs in Pedigree Analysis: Applications to Haplotyping, Location Scores, and Marker-Sharing Statistics

Eric Sobel\(^1\) and Kenneth Lange\(^2\)

\(^1\)Department of Genetics, Stanford University, Stanford; and \(^2\)Departments of Biostatistics and Mathematics, University of Michigan, Ann Arbor

Summary

The introduction of stochastic methods in pedigree analysis has enabled geneticists to tackle computations intractable by standard deterministic methods. Until now these stochastic techniques have worked by running a Markov chain on the set of genetic descent states of a pedigree. Each descent state specifies the paths of gene flow in the pedigree and the founder alleles dropped down each path. The current paper follows up on a suggestion by Elizabeth Thompson that genetic descent graphs offer a more appropriate space for executing a Markov chain. A descent graph specifies the paths of gene flow but not the particular founder alleles traveling down the paths. This paper explores algorithms for implementing Thompson’s suggestion for codominant markers in the context of automatic haplotyping, estimating location scores, and computing gene-clustering statistics for robust linkage analysis. Realistic numerical examples demonstrate the feasibility of the algorithms.

1. Introduction

Linkage calculations on human pedigree data are routinely done in thousands of genetics laboratories around the world. These calculations are some of the most computationally demanding tasks in modern biology. If anything, the computational bottlenecks are intensifying because of the shift from RFLP markers to the more polymorphic short tandem repeat (STR) markers. One possible remedy for these bottlenecks is to resort to stochastic methods for likelihood computation. During the past few years, mathematical and statistical geneticists have imported and extended methods such as Gibbs sampling (Geman and Geman 1984), the Metropolis algorithm (Metropolis et al. 1953), and data augmentation (Tanner 1991). In particular, Gibbs sampling, the Metropolis algorithm, or both have been tried in linkage analysis (Lange and Matthyse 1989; Lange and Sobel 1991; Guo and Thompson 1992; Thomas and Cortessis 1992; Sobel and Lange 1993), segregation analysis (Guo and Thompson 1992), the mixed model (Guo and Thompson 1994), variance components models (Guo and Thompson 1991; Sobel and Lange 1993), and haplotyping (Sobel et al. 1995).

In spite of these advances, stochastic methods have not penetrated genetic practice as deeply as some of us originally anticipated. Although these methods can give good approximate solutions to previously intractable problems, computing times are unacceptably long, and user-friendly software is largely unavailable. In a recent series of papers, Thompson has suggested a novel device for improving the accuracy and speed of the stochastic methods (Thompson 1994a, 1994b, and in press). Previous theory involved simulations conducted on the space of genetic descent states for a pedigree. Roughly speaking, a genetic descent state specifies the paths of gene flow in a pedigree and the actual founder alleles dropped down each path. Thompson argues that the much smaller space of genetic descent graphs offers a better space on which to conduct simulations. A genetic descent graph includes the gene flow information contained in a genetic descent state but omits specification of the actual alleles dropped down each path.

As motivation for preferring genetic descent graphs to genetic descent states, consider the two pedigrees depicted in figure 1. These pedigrees manifest problems seen in larger pedigrees. (Throughout this paper we list ordered genotypes as maternal allele/paternal allele, e.g., 1/2. Phenotypes at a codominant marker locus we list as unordered genotypes in braces, e.g., \{1, 2\}.) For the pedigree in part a of figure 1, a random walk on the set of genetic descent states will spend an inordinate amount of time trying out possible alleles for the unknown founder gene labeled by a question mark (?). Unless this gene is allele 2, it cannot be passed to either child, and repeated sampling of alternative alleles is wasted effort. Indeed, in haplotyping, the best allele is obviously either allele 2 or the most frequent allele, and in computing location scores and clustering statistics, all non-2 alleles are equivalent. Part b shows two genetic-descent states,
A and B, in which the only change is that every ordered genotype has been reversed. These two states have identical likelihoods and should in some sense be considered identical. However, a random walk on genetic descent states must execute many elementary rearrangements to pass from one state to the other. A random walk on genetic descent graphs appropriately ignores the distinctions between these two states.

In the current paper, we introduce detailed algorithms that make the space of genetic descent graphs a good vehicle for stochastic simulation on pedigrees. For codominant markers we show how to compute efficiently the likelihood of any genetic descent graph compatible with marker phenotypes. We also show how to solve the haplotyping problem by finding the most likely genetic descent state compatible with marker phenotypes. These algorithms flesh out the Markov chain Monte Carlo (MCMC) method devised by Thompson (Thompson 1994a, 1994b, and in press). Our theoretical advances are illustrated by applications to automatic haplotyping, estimating location scores, and computing statistics indicative of marker allele sharing among pedigree members affected by the same disease. The specific statistics introduced for allele sharing should prove useful in conducting robust linkage analyses of common diseases based on marker clustering among affected.

2. Definitions

Our previous expositions of the Metropolis-coupled MCMC method for pedigrees emphasized genetic descent states (Lange and Matthyssen 1989; Lange and Sobel 1991; Sobel and Lange 1993; Sobel et al. 1995). To explain the notion of a genetic descent state in more detail, suppose that we observe marker phenotypes at $l$ loci of a pedigree with $p$ people of whom $f$ are founders. A genetic descent state in this setting completely specifies the gene flow within the pedigree at these loci. Gene flow can be separated into two parts: the paths that the founder genes take as they descend through the pedigree and the allelic form assumed by each founder gene. Figure 2 exhibits a pedigree with ordered genotypes at a single locus. Part $a$ is the conventional representation of the pedigree data; part $b$ shows a more informative genetic descent state that is one of the four compatible with the ordered genotype data.

The paths of gene flow in a genetic descent state constitute a genetic descent graph. This directed graph possesses $2lp$ nodes, each of which is a particular combination of locus, person, and source. If we think of a gene occupying each node, then the source of the node tells us whether the gene is maternal or paternal in origin. The arcs of a genetic descent graph connect a parental node to a descendant child node at the same locus and determine whether a parent contributes a grandmaternal or a grandpaternal gene to the child at the locus. Pedigree $b$ of figure 2 provides an example of a genetic descent graph if one ignores the listed founder alleles.

Rooted at each founder node there is a directed tree incorporating exactly those nodes in the genetic descent graph that inherit the corresponding founder gene. This genetic descent tree forms a connected component of the genetic descent graph. In all there are $2lf$ genetic

![Figure 1](image1.png)

**Figure 1**  Two examples illustrating why genetic descent graphs are preferable to genetic descent states.

![Figure 2](image2.png)

**Figure 2**  Two representations of a pedigree with ordered genotypes at a single locus. $a$, Conventional representation. $b$, More informative genetic descent state representation with gene flow patterns and founder alleles fully specified.
Sobel and Lange: Descent Graphs in Pedigree Analysis

Figure 3  a, Genetic descent graph in which each descent tree is labeled above its founder gene. Known phenotypes for this pedigree are shown in braces. b, Founder-tree graph corresponding to this descent graph and phenotyping pattern. c, Table of the legal candidate allele vectors for the components of the founder-tree graph.

descent trees. By definition, the nodes of a genetic descent tree all belong to the same locus. For the sake of brevity, in the sequel we will omit the modifier "genetic" from "genetic descent tree," "genetic descent graph," and "genetic descent state."

Finally, it is convenient to proceed to one higher level of abstraction and define a founder-tree graph corresponding to each descent graph. The nodes of this affiliated undirected graph are the descent trees of the descent graph. Two nodes are connected by an edge if and only if the two corresponding descent trees pass through the same typed locus of some person in the pedigree. This definition precludes connecting two descent trees associated with different loci. Part a of figure 3 shows a descent graph in which each descent tree is labeled above its rooting founder gene. Part b of figure 3 shows the founder-tree graph corresponding to this descent graph, assuming all nonfounders and no founders are typed at the locus. It may be that two descent trees at the same locus mutually impinge on more than one person typed at the locus. Although this information is relevant to discerning whether the two trees are genetically compatible with the observed phenotypes in the pedigree, for the sake of simplicity we still view the descent trees as connected by a single edge. If a descent tree intersects no one typed at its associated locus, then the descent tree is isolated from all other descent trees in the founder-tree graph.

3. Candidate Alleles for a Descent Tree

As noted above, a descent state specifies the allele type of each founder gene. In general we will consider only descent states compatible with the observed marker types within the pedigree. There may be a multitude of such legal descent states consistent with a given descent graph. The typed people within the pedigree put constraints on the number of legal descent states. If we imagine assigning an allele type to each descent tree via the founder gene at its root, then the fates of two connected descent trees are coupled together by the common typed people through which they pass. For instance, if they both pass through an individual having unordered genotype \(a_i, a_j\), then one of the descent trees must carry allele \(a_i\) and the other allele \(a_j\). They cannot produce a legal descent state if they both carry allele \(a_i\) or allele \(a_j\) or if one carries a completely different allele. Refinement of these simple ideas involving the founder-tree graph will permit us to write the joint likelihood of a descent graph and the observed marker phenotypes in the pedigree.

One can subdivide the nodes of the founder-tree graph into connected components. These components are sets of descent trees and should not be confused with the components of the descent graph, which are single descent trees. In the founder-tree graph, two nodes belong to the same component if and only if one can travel from one node to the other by a finite sequence of edges. A component is said to be singleton if it consists of a single node. In a nonconsanguineous pedigree, a descent tree forms a singleton component if it passes through no one typed at its associated locus. In this case, all alleles are legal for the founder gene of the descent tree. In a consanguineous pedigree, a descent tree can still form a singleton component if it descends through a typed child via both parents. If the typed child is homozygous with genotype \(a_i/a_i\), then \(a_i\) is the only allele permitted for the founder gene. If the typed child is heterozygous, then no legal allele exists for the founder gene.

The situation for a multinode component of the founder-tree graph is equally simple. If we label the nodes of the component as \(i_1, \ldots, i_k\), then the founder gene for node \(i_1\) is transmitted to some typed person who is either homozygous or heterozygous. If the person is homozygous, then there is only one legal choice for founder gene \(i_1\). Because founder gene \(i_1\) is connected to another founder gene through this typed person or another typed person, the connected founder gene is also completely determined. This second founder gene is in
turn connected to a third founder gene through some typed person. Hence, the third founder gene is also uniquely determined. In general, a cascade of connecting edges completely determines the permissible alleles for each of the founder genes of the component, unless of course an inconsistency is encountered at some step. If founder gene \( i_1 \) passes through a typed heterozygote, then \( i_1 \) may take the form of either participating allele. Once one of these two alleles is chosen for \( i_1 \), then the allelic form of all other founder genes in the component is determined by the argument just given. Thus, we can summarize the situation for a multidnode component \( i_1, \ldots, i_k \) by noting that either two, one, or no allele vectors \( a = (a_{i_1}, \ldots, a_{i_k}) \) can be legally assigned to the founder genes of the component. Part \( c \) of figure 3 shows all the legal candidate vectors for the components of the founder-tree graph shown in part \( b \) and the phenotyping shown in part \( a \) of that figure.

4. Likelihood of a Descent Graph

The joint likelihood of a descent graph \( \hat{G} \) and a marker phenotype vector \( M \) can be written as a sum of the joint likelihoods of \( M \) and all descent states \( G \) consistent with the gene flow specified by \( \hat{G} \). In symbols,

\[
Pr(\hat{G} \cap M) = \sum_{G \rightarrow \hat{G} \cap M} Pr(G), \tag{1}
\]

where \( G \rightarrow \hat{G} \cap M \) denotes consistency between \( G \) and both \( \hat{G} \) and \( M \). Under Hardy-Weinberg and linkage equilibrium for linked codominant markers, the probability \( Pr(G) \) reduces to a product of the founder allele frequencies involved in the descent state \( G \), relevant powers of \( p_i \), and the recombination fractions and their complements for the adjacent intervals separating the markers (Lange and Sobel 1991; Sobel and Lange 1993). Let us designate the allele frequency factor as \( Prior(G) \) and the transmission factor as \( Trans(G) \). Because all compatible descent states \( G \rightarrow \hat{G} \) exhibit the same transmission pattern, \( Trans(G) \) depends only on the descent graph \( \hat{G} \) and not on the particular representative chosen from the set \( \{G : G \rightarrow \hat{G} \cap M\} \). Hence, we can re-express the likelihood (1) as

\[
Pr(\hat{G} \cap M) = Trans(\hat{G}) \sum_{G \rightarrow \hat{G} \cap M} Prior(G). \tag{2}
\]

To simplify formula (2) further, label the connected components of the founder tree graph \( C_1, \ldots, C_m \). Then for a given consistent descent state \( G \rightarrow \hat{G} \cap M \), let \( a \) be the vector of alleles assigned to component \( C_i \). Under conditions of genetic equilibrium, each founder gene is sampled independently; therefore,

\[
Prior(G) = \prod_{i=1}^{m} Pr(a_i), \tag{3}
\]

where

\[
Pr(a_i) = \prod_{i} Pr(a_{i}),
\]

the allele vector \( a_i \) having constituent alleles \( a_{i_k} \). By construction, the founder genes assigned to different components do not impinge on one another. In other words, the set of founder genes consistent with \( \hat{G} \) and \( M \) is drawn from the Cartesian product of the sets \( S_1, \ldots, S_m \) of legal allele vectors for the components \( C_1, \ldots, C_m \), respectively. Applying the distributive rule to equation (3) consequently yields

\[
\sum_{G \rightarrow \hat{G} \cap M} Prior(G) = \prod_{i=1}^{m} Pr(C_i), \tag{4}
\]

where

\[
Pr(C_i) = \sum_{a \in S_i} Pr(a_i).
\]

As mentioned in the previous section, a set \( S_i \) either contains all allele vectors or just two, one, or none. In the first case, \( \sum_{a \in S_i} Pr(a_i) = 1 \), and in the last case \( \sum_{a \in S_i} Pr(a_i) = 0 \). In the remaining two cases, the product formula \( Pr(a_i) = \prod_{j} Pr(a_{j_k}) \) is applicable. Hence, calculation of \( \sum_{G \rightarrow \hat{G} \cap M} Prior(G) \) reduces to easy component-by-component calculations.

For easy reference, we combine formulas (2) and (4) into the single formula

\[
Pr(\hat{G} \cap M) = Trans(\hat{G}) \prod_{i=1}^{m} Pr(C_i). \tag{5}
\]

After we have introduced a Markov chain on descent graphs, we will demonstrate how to update the likelihood (5) quickly as the chain steps from descent graph to descent graph.

5. Computing Location Scores

Location scores are used to position a trait locus relative to an existing set of mapped markers (Lange and Sobel 1991; Lathrop et al. 1983). Suppose the unknown trait position is denoted by \( d \), the trait phenotypes for a pedigree by \( T \), and the marker phenotypes by \( M \). The information necessary to plot a location score is contained in the conditional probability \( Pr(d | M) \). If we can sample from marker descent graphs \( \hat{G} \) given the marker types \( M \), then we can employ standard pedigree likelihood programs such as MENDEL (Lange et al.
1988) to estimate $Pr_d(T \mid M)$. The basis for this computation is the obvious decomposition

$$Pr_d(T \mid M) = \sum_G Pr_d(T \mid G)Pr(G \mid M), \quad (6)$$

which relies implicitly on the assumption of linkage equilibrium between the trait and marker loci. To evaluate equation (6) we run a Markov chain on descent graphs $G$. This chain has equilibrium distribution matching the conditional distribution $Pr(G \mid M)$. If a sequence of descent graphs $G_1, \ldots, G_n$ is generated by running the chain, then the sample average

$$\frac{1}{n} \sum_{i=1}^n Pr_d(T \mid G_i)$$

will approximate $Pr_d(T \mid M)$ well for $n$ sufficiently large.

Deterministic computation of $Pr_d(T \mid G)$ can be achieved by reprogramming MENDEL to recognize a descent graph $G$ at the marker loci as legitimate input. This entails conveying to MENDEL an ordered pair of source indicators at each marker locus of each non-founder. For instance, when the ordered pair of source indicators $2/1$ is input to the appropriate version of MENDEL as a marker phenotype, then the program interprets the maternal or left gene of the given person as having a grandpaternal source—labeled 2—and the paternal or right gene of the person as having a grandmaternal source—labeled 1. For this subterfuge to work, the apparent genetic inconsistencies introduced by treating source pairs as ordered genotypes must be handled by shutting down MENDEL's automatic consistency checks. Also, MENDEL’s routine for computing gamete transmission probabilities must be altered to deal with source information rather than ordered genotypes. Finally, MENDEL’s calculation of founder haplotype frequencies must be modified to ignore allele frequencies at the marker loci. With these changes to the user portion of MENDEL, the conditional likelihood $Pr_d(T \mid G)$ can be computed by dividing the likelihood calculated by MENDEL, $Pr_d(T \cap G)$, by the transmission probability $Trans(G)$. Note that marker allele frequencies are omitted from calculation of both the numerator and denominator of $Pr_d(T \mid G)$. Furthermore, since the trait locus is usually biallelic, and since sampling from the Markov chain fills in all of the missing information on marker gene flow, deterministic likelihood calculation is generally quick.

6. The Descent Graph Markov Chain

The collection of descent graphs over a pedigree becomes a finite Markov chain if we incorporate transition rules for moving between descent graphs. Thompson (Thompson 1994a, 1994b, and in press) has suggested the simple move of switching the origin of an arc descending from a parent to a child, from the parental maternal node to the parental paternal node, or vice versa. The arbitrary child node chosen we will term the “pivot node.” It is comprised of a nonfounder, a locus, and a source. This source switch, which we refer to as transition rule $T_0$, is also one of the two basic transition rules for moving between descent states discussed in Lange and Matthysse (1989), Lange and Sobel (1991), and Sobel and Lange (1993). The other basic rule mandates an allele change for a founder gene. When descent graphs are substituted for descent states, founder allele substitution is no longer relevant. Figure 4 illustrates the source switch rule $T_0$.

In our previous work on Markov chains over descent states (Lange and Matthysse 1989; Lange and Sobel 1991; Sobel and Lange 1993), we included not only the two basic transition rules mentioned above, but also two composite rules designed to speed up the circulation of the chain. Composite rules constructed from several coordinated source switches are also useful for moving between descent graphs. Our first composite transition rule $T_1$ is illustrated in figure 5. The idea behind $T_1$ is simple. At a given person and locus we can define two descent subtrees, one rooted at the maternal node and the other rooted at the paternal node. These two subtrees are detached from their rooting nodes and rerooted at the opposite nodes. More formally, transition rule $T_1$ begins by choosing a person $i$ and a locus $l$. Next, a $T_0$ transition is performed at the nodes determined by each child of $i$, the given locus $l$, and the sex of $i$. Thus every child of $i$ who previously inherited $i$’s maternal gene will now inherit $i$’s paternal gene, and vice versa.

The second composite transition rule on descent graphs has two variants $T_{2a}$ and $T_{2b}$. These are illustrated in figure 6. Each variant begins by choosing a locus $l$ and a pivot couple $i$ and $j$ with common children. A total of four descent subtrees are rooted at the parents $i$ and $j$. In rule $T_{2a}$, the subtree rooted at the maternal

![Figure 4](image-url)  

**Figure 4** Example of transition rule $T_0$, a source switch. The solid-black circle indicates the selected pivot node. Only a single locus is shown. The thick vertical arc indicates the source specification that changes.
a single black circle indicates Figure 6 trees, we have this process rooted subtree the maternally rooted either parent of maternal node spouse are descending through relevant subtrees parent's nal nodes, these forbidden of node that change. be made is locus shown. The pivot person selected for this transition. Only a single locus is shown.

node of \( i \) is exchanged with the subtree rooted at the maternal node of \( j \); likewise, the paternally rooted subtrees of \( i \) and \( j \) are exchanged. In rule \( T_{2b} \) we exchange the maternally rooted subtree of \( i \) with the paternally rooted subtree of \( j \) and vice versa. Two subtle points of this process are worth stressing. After swapping subtrees, we have paternally derived genes flowing to maternal nodes, and vice versa. The obvious adjustments must be made in the children and grandchildren to correct these forbidden patterns of gene flow. Furthermore, if either parent has children with another spouse, then that parent's relevant subtrees are reduced. Only the paths descending through the children shared with the chosen spouse are now pertinent. The formal sequence of \( T_0 \) and \( T_1 \) transitions invoked to execute a \( T_{2a} \) or \( T_{2b} \) transition is explained in the appendix.

One of the complications in constructing a Markov chain on legal descent states is that two states may not communicate in the presence of three or more alleles per marker (Lange and Sobel 1991). This hazard persists when we pass to the set of legal descent graphs as the underlying space of the Markov chain. Figure 7 gives a counterexample involving a single marker locus. In the pedigree depicted in figure 7, all founders are typed and homozygous; the great-grandchild is typed and heterozygous. This great-grandchild must receive his allele 1 from one pair of great-grandparents and his allele 2 from the other pair. The two possibilities are labeled descent graph A and descent graph C. However, it is impossible to move in a finite number of transitions from descent graph A to descent graph C without passing through an illegal descent graph such as descent graph B, where the great-grandchild inherits a homozygous genotype.

The remedy to this dilemma is to "tunnel through" illegal descent graphs by taking multiple transitions per step of the Markov chain (Sobel and Lange 1993). In practice, we employ a random number of transitions per step of the chain. This procedure permits the chain to pass through illegal descent graphs on its way between
legal descent graphs. Among the many devices for selecting the random number $K$ of transitions per step, we prefer random sampling from a geometric distribution with mean 2. This procedure entails taking a single transition with probability $\frac{1}{2}$, two transitions with probability $\frac{1}{4}$, three transitions with probability $\frac{1}{8}$, and so forth. For each transition one randomly selects a transition rule and a pivot person and locus. If the transition rule selected is $T_0$, then one also randomly selects a maternal or paternal node to switch. If one of the $T_2$ transitions is selected, then one also randomly selects a spouse of the pivot person.

Although random, not all selections are made uniformly. For example, for nonfounders we select transition $T_0$ more often than $T_1$ and $T_1$ more often than $T_2$. Other choices, such as maternal-versus-paternal node for a $T_0$ transition or spouse for a $T_2$ transition, are made uniformly. To increase the likelihood that the random walk takes large steps, we oversample certain people and loci. Some examples of oversampling are (1) giving people untyped at a locus greater weight when pivoting on that locus; and (2) after the initial transition in a step, giving neighboring loci of a just-selected locus greater weight. Here, a neighboring locus of $l$ is $l$ itself or one of the loci flanking $l$. To maintain reversibility of our Markov chain, it is crucial that within a step independent choices are made for both the sequence of transition rules invoked and the sequence of people and sources to which these transitions are applied. In the appendix this requirement is discussed further in the context of the algebraic structure of the set of transitions.

To construct a Markov chain using these transition rules and possessing the correct equilibrium distribution, one can employ the Metropolis mechanism (Metropolis et al. 1953; Hastings 1970). The Metropolis mechanism divides a step into a proposal stage and an acceptance stage. In the proposal stage the number and type of transitions and the pivot nodes are chosen. Let

$$q_{ij} = \Pr(\hat{G}_j \text{ is the proposed next graph | } \hat{G}_i \text{ is the current graph})$$

be the proposal probability for moving from descent graph $\hat{G}_i$ to descent graph $\hat{G}_j$. We show in the appendix that the proposal matrix $Q = (q_{ij})$ is symmetric. Because a single step can consist of an unlimited number of transitions, it is possible to move from any legal descent graph to any other legal descent graph in a single step. Thus, all entries of $Q$, including its diagonal entries, are positive.

Once a step from $\hat{G}_i$ to $\hat{G}_j$ is proposed, it is accepted with the Metropolis probability (Metropolis et al. 1953)

$$a_{ij} = \min\left\{1, \frac{\Pr(\hat{G}_j \mid M)}{\Pr(\hat{G}_i \mid M)}\right\} = \min\left\{1, \frac{\Pr(\hat{G}_i \cap M)}{\Pr(\hat{G}_j \cap M)}\right\}.$$ 

On the basis of this criterion, a more likely descent graph is always accepted, and an illegal descent graph is always rejected. A less likely but still legal descent graph is sometimes accepted and sometimes rejected.

The overall probability of moving from $\hat{G}_i$ to $\hat{G}_j$ is

$$p_{ij} = \begin{cases} q_{ij}a_{ij} & \text{if } j \neq i, \\ 1 - \sum_{k \neq i} p_{ik} & \text{if } j = i. \end{cases}$$

In view of the positivity of the proposal matrix $Q$, the final single-step matrix $P = (p_{ij})$ is also positive on the set of legal descent graphs. This fact implies that the Markov chain on legal descent graphs is irreducible and aperiodic. Finally, the usual arguments (Lange and Sobel 1991; Sobel and Lange 1993) involving reversibility and detailed balance demonstrate that the Markov chain has equilibrium distribution given by the conditional probabilities $\Pr(\hat{G} \mid M)$.

7. Implementation of the Markov Chain

The above broad outline of the Markov chain omits several crucial issues. For instance, one needs to find a legal starting state for the chain. Sobel et al. (1995) describe an iterative, genotype-elimination algorithm that solves this problem locus-by-locus. This algorithm produces a legal, ordered genotype vector $G^*$ that can be easily transformed into a legal descent state $G$ and a corresponding legal descent graph $\hat{G}$. In general, one should not settle for simply any legal descent graph. Our experience with computing location scores via descent states suggests that starting with a nearly optimal descent graph is wise. Although all legal descent graphs communicate in theory, in practice there may be nearly insurmountable barriers to communication, and the best results are apt to be achieved by initiating sampling of the conditional probability space of descent graphs in the vicinity of its dominant peak. Of course, finding the dominant peak is closely related to finding the most likely marker haplotype vector for the underlying pedigree. This topic will be discussed in the next section.

Once the Markov chain is underway, fast updating of the likelihood (5) is vitally important. It suffices to consider a proposed descent graph $\hat{G}_i$ differing by a single source switch, transition $T_0$, from the current descent graph $\hat{G}_j$. Under Haldane’s model of independent recombination on disjoint intervals, there are two cases to consider in updating the transmission probability $\text{Trans}(\hat{G}_j) \rightarrow \text{Trans}(\hat{G}_i)$. If the source switch occurs at the first or last marker locus, then $\text{Trans}(\hat{G}_i)$ is computed from $\text{Trans}(\hat{G}_j)$ by substituting the flanking recombination fraction $\theta$ for its complement $1 - \theta$, or vice versa. If the source switch occurs at an internal marker loci,
then there are two flanking recombination fractions and two corresponding substitutions.

Updating of \( \prod_{i=1}^{m} \text{Pr}(C_i) \) in equation (5) presents a greater challenge. However, considerable simplification can be achieved by noting that for a single source switch the necessary changes can be isolated to at most two components of the founder-tree graph. In effect a source switch detaches part of one descent tree and attaches the detached part to another descent tree. See figure 4 for an example. This action deletes some edges of the founder-tree graph and adds other edges. If the first descent tree belongs to component \( C_1 \), then the second descent tree may belong to \( C_1 \) or to a second component \( C_2 \). In the former case, \( C_1 \) may be split by the source switch into two components \( C_1 \setminus D \) and \( D \). In the latter case, the source switch transfers some nodes from \( C_1 \) to \( C_2 \) to form reconfigured components \( C_1 \setminus D \) and \( C_2 \cup D \). Depending on the initial descent graph \( \hat{G}_1 \), the set \( D \) of transferred nodes can range from \( C_1 \) itself to the empty set.

Formula (5) highlights the importance of these observations. Because most of the prior factors remain intact, the likelihood need only be multiplied by either the ratio \( \frac{\text{Pr}(C_1 \setminus D) \text{Pr}(C_2 \cup D)}{\text{Pr}(C_1 \setminus D) \text{Pr}(C_2)} \) or the ratio

\[
\frac{\text{Pr}(C_1 \setminus D) \text{Pr}(C_2 \cup D)}{\text{Pr}(C_1) \text{Pr}(C_2)} ,
\]

to account for changes in permitted founder genes in passing from descent graph \( \hat{G}_i \) to descent graph \( \hat{G}_j \). If additional transitions beyond a single source switch are undertaken, then matters become slightly more complicated, but the same principles apply. It is only fair to add that there is an inevitable overhead levied by the bookkeeping tasks of keeping track of the connected components of the founder-tree graph. We omit these tedious details because they are so closely tied to actual computer code.

8. Simulated Annealing and Haplotype Construction

We have already mentioned the desirability of starting the Markov chain in a descent graph with nearly maximal likelihood. This favorable state of affairs can be achieved by simulated annealing beginning with any legal descent graph. Simulated annealing employs the Markov chain with an acceptance probability

\[
a_{ij} = \min \left\{ 1, \left[ \frac{\text{Pr}(\hat{G}_i \cap M)}{\text{Pr}(\hat{G}_j \cap M)} \right]^{\frac{1}{\beta}} \right\} ,
\]
depending on a temperature parameter \( \beta \) that is slowly sent to zero (Kirkpatrick et al. 1983; Press et al. 1992). In the early stages of simulated annealing, many unfavorable steps are taken, and the space of legal descent graphs is broadly sampled. As the temperature \( \beta \to 0 \), fewer and fewer unfavorable steps are taken, and the annealing process eventually settles on some nearly optimal descent graph. This behavior is consistent with the behavior of simulated annealing on a variety of combinatorial optimization problems. Of course, simulated annealing is not infallible, and multiple independent runs are almost always advisable.

Simulated annealing is also capable of finding an optimal or a nearly optimal legal descent state for a pedigree. Sobel et al. (1995) solve this "haplotyping" problem by conducting simulated annealing directly on the space of legal descent states. Here we indicate how to use the much smaller space of legal descent graphs. The essence of our new solution is that we assign to each descent graph the likelihood of its most likely descent state. This negative "energy" function can then be used in a simulated annealing run over the space of descent graphs to identify the descent graph with least energy and therefore the most likely descent state.

A best descent state corresponding to the descent graph \( \hat{G} \) maximizes the product

\[
\text{Prior}(G) = \prod_{i=1}^{m} \text{Pr}(a_i) ,
\]

where \( a_i \) is any legal allele vector assigned to component \( C_i \) of the founder tree graph associated with \( \hat{G} \). To maximize \( \text{Prior}(G) \), one simply maximizes each factor \( \text{Pr}(a_i) \) over its set \( S_i \) of possible allele vectors. When the set \( S_i \) has one or two members, then it is trivial to choose the best member. If \( S_i \) consists of more than two members, then \( C_i \) reduces to a single descent tree, and \( S_i \) contains all possible alleles for the corresponding founder gene. In this case, one chooses the allele with maximum population frequency. Again, it should be pointed out that in executing a step of the simulated annealing chain, at most two components of the founder tree graph are changed by a \( T_0 \) or a \( T_1 \) transition and at most four by a \( T_2 \) transition. These limits simplify energy assignment in the same way that they simplify likelihood computation. Also, since reversibility of the Markov chain has no bearing on simulated annealing, we can now use correlated choices of people and transition rules within a step. For example, after the initial transition in a step, we give neighboring people of the immediately preceding pivot person greater weight. A neighboring person of \( i \) is \( i \) himself or one of his parents, spouses, children, or siblings.

Our experience using simulated annealing to find a nearly optimal initial descent state for location score analysis suggests that including trait information can be very helpful. On the basis of marker phenotypes alone, two or more descent states may have similar likelihoods. However, once the trait is included—usually midway
in each possible marker interval in succession—the candidate states can yield sharply different likelihoods. This phenomenon carries over to descent graphs. Of course, if the trait locus is not codominant, then our methods for computing descent graph likelihoods fail but not our previous methods for computing descent state likelihoods. Hence a good compromise might be to conduct simulated annealing on a combination of the descent graphs at the marker loci and on the descent states of the trait locus.

9. Statistics for Marker Gene Clustering

Markov chain sampling of descent graphs also permits evaluation of marker allele sharing among pedigree members affected by the same disease. For epidemiological purposes, the natural null hypothesis is that disease transmission is independent of marker transmission. When independence fails, one expects to see a clustering among affecteds of a few marker genes descending from the pedigree founders. Some interesting statistics based on single-locus descent graphs are:

(A) The total number of different descent trees contributing to the marker genes appearing among affecteds.

(B) The maximum number of marker genes among affecteds attributable to any one descent tree.

(C) The entropy of the marker genes among the affecteds as measured by the statistic

$$\varepsilon = -\sum_{k=1}^{2^f} \frac{n_k}{2n} \ln \frac{n_k}{2n}.$$ 

Here, \(f\) is the number of founders, \(n\) is the number of affecteds, and \(n_k\) is the number of marker genes among the affecteds attributable to descent tree \(k\).

(D) The extent of gene sharing among affected pairs \((i, j)\) as measured by \(\sum_{i,j} Z_{ij}\), the sum of their kinship coefficients conditional on the current descent graph. If \(i_{ml}/i_p\) and \(j_{ml}/j_p\) are the ordered genotypes of \(i\) and \(j\), then the pairwise statistic \(Z_{ij}\) is defined by

$$Z_{ij} = \frac{1}{4} \left[ 1_{i_{ml}=i_p} + 1_{i_{ml}=j_p} + 1_{j_{ml}=i_p} + 1_{j_{ml}=j_p} \right],$$

where equality between two genes indicates identity by descent.

For the statistics (A) and (C), a small value suggests clustering, while for statistics (B) and (D) the opposite is true. Statistics similar to our kinship statistic (D) have previously been explored by Weeks and Lange (1988) and Whittemore and Halpern (1994). Statistic (A) is apt to be the most powerful in detecting linkage to a recessive disease. Statistic (B) is apt to be the most powerful for a dominant disease. Statistics (C) and (D) are generic statistics indicating whether there are a few descent trees that are overly represented among the affecteds’ marker genes. We have chosen entropy rather than its negative, information, to avoid negative values for statistic (C).

Although none of the four statistics can be evaluated directly from pedigree phenotypes, all can be trivially evaluated from a descent graph. To reduce the noise associated with random sampling of a single descent graph, we propose approximating the conditional expectation of each statistic, given marker phenotypes and the constellation of affecteds within a pedigree. Good approximations to these conditional expectations can be found by running our Markov chain on descent graphs and computing the corresponding sample means.

In symbols, if \(S\) is any one of these statistics and \(M\) is the observed marker data, then we estimate \(R = E(S|M)\) by \(\frac{1}{m} \sum_{r=1}^{m} S_r\), where \(S_r\) is the value of the statistic \(S\) at epoch \(r\) of \(m\) sampled epochs of the Markov chain. For purposes of discussion, we assume that \(m\) is large enough so that \(R\) is estimated without error.

Obviously, application of these test statistics raises interesting inference questions. We suggest viewing the statistics as exploratory tools and comparing their values to the unconditional distributions of the same statistics ignoring marker phenotypes in the pedigree. Simulating the unconditional distributions can be accomplished by “gene dropping” techniques and does not require running a complicated Markov chain. The identity \(E(R) = E(S)\) holds because the expectation of a conditional expectation is an ordinary expectation; consequently, the mean of \(R\) can be approximated with good precision by the sample mean of the gene-dropping simulation. For the kinship-sharing statistic (D), we can compute the mean \(E(S)\) exactly. The variance of \(S\) satisfies

$$\text{Var}(S) = \text{Var}(E(S|M)) + E(\text{Var}(S|M))$$

$$= \text{Var}(R) + E(\text{Var}(S|M)).$$

The now obvious inequality \(\text{Var}(R) \leq \text{Var}(S)\) suggests that empirical \(P\) values obtained from a simulation of \(S\) should generally be conservative estimates of the true \(P\) values for statistic \(R\).

Combining statistics from different pedigrees also poses a problem. The most natural tactic is to sum the values from different pedigrees. This will probably work well for the first three statistics, but for the kinship statistic it puts too much weight on pedigrees with many affecteds. Some reweighting to correct for the quadratic growth in the number of affected pairs is indicated. For want of more compelling arguments than those given by Weeks and Lange (1988) and Whittemore and Halpern (1994), we use the weight \(\sqrt{2/[n(n-1)]}\) for a pedigree with \(n\) affecteds.
10. Testing the Methods

We now turn to some results on haplotyping, location scores, and clustering statistics from both real and simulated data. The first pedigree we examine is from a study on episodic ataxia (EA) by Litt et al. (1994). With their permission, we comment on an enlarged and slightly corrected version of their pedigree 4. This pedigree now contains 29 people, 27 of whom are at least partially typed at 9 highly polymorphic chromosome 12p markers. Table 3 of Litt et al. (1994) provides a map of these markers derived from linkage analysis of CEPH families (Dausset et al. 1990). Litt et al. reject the CEPH map because it "would result in an obligate triple crossover, within a 3-cM region, in individual 113" (Litt et al. 1994, p. 706). Accordingly, their figure 2A presents a haplotype vector for the pedigree using the alternative order that shifts locus D12S99 two positions distal to its CEPH position. They claim that this alternative order reduces the apparent triple crossover to a single crossover.

All of our haplotype reconstructions for this pedigree using the descent graph method agree in essential details with the haplotype reconstructions previously obtained by using the descent state method (Sobel et al. 1995). Figure 8 displays a representative haplotype vector found under the original CEPH order of the markers. We find no triple crossover in this haplotype vector. In fact, the haplotype vector eliminates three superfluous recombination events postulated in the Litt et al. reconstruction. Our haplotype revisions fortunately do not affect the conclusion drawn by Litt et al. that the EA locus lies between D12S372 and the cluster pY21/1-pY22/1-KCNA5-D12S99. It is noteworthy that under the Litt et al. marker map our best guess of the haplotype vector remains unchanged except for the necessary reordering of loci along each haplotype.

To evaluate the haplotyping methods in a controlled setting, we simulated a haplotype vector for a consanguineous pedigree segregating the recessive disease ataxia-telangiectasia (A-T) on chromosome 11q (Ziv et al. 1992; Lange et al. 1995; Savitsky et al. 1995). This real pedigree, shown in figure 9, has 12 people in 4 generations. To make our simulation as realistic as possible, we employed 17 highly polymorphic, codominant markers separated by interlocus recombination fractions ranging from .0015 to .065. The simulated haplotype vector for this pedigree contains 8 recombination events and has a log likelihood of -116.16 as computed by MENDEL. From the simulated haplotype vector we created two data sets. The first, AT.1, consists of the pedigree with all haplotype pairs reduced to their corresponding phenotypes. In the real pedigree, individuals 2, 3, and 4 are untyped; this missing data pattern is imposed on the simulated data set AT.1 to produce data set AT.2.

For the AT.1 data, the haplotype vectors found by all runs of our haplotyping program are essentially identical to the simulated haplotype vector. All shared the same log likelihood and recombination count. This was also the case for results from the descent state method (Sobel et al. 1995). For the AT.2 data, all runs found haplotype vectors more likely than the simulated haplotype vector. Two of 10 runs gave a haplotype vector with six recombination events and a log likelihood of -110.52; the remaining 8 runs gave a haplotype vector with only five recombination events and a log likelihood of -109.59. These results are considerably better than those of the descent state method (Sobel et al. 1995). In that case the best result—found in only a third of the runs—showed 10 recombinants and a log likelihood of -115.76.

One haplotyping run for the AT.1 data takes 19:37 (minutes:seconds) on a 225 MHz Alpha central processing unit (CPU) and 23:07 on a 120 MHz Pentium CPU. For the AT.2 data the run time is 22:07 on the Alpha and 27:11 on the Pentium. The descent state version took between 9 and 10 min for each run on the Alpha (Sobel et al. 1995). We note that missing data have little impact on overall computation times for either method on these and similar problems.

We also used the EA pedigree of Litt et al. (1994) for testing the location score algorithms. To compute exact location scores via MENDEL we were forced to eliminate the three loci pY21/1, KCNA5, and D12S99 from the four-locus cluster mentioned above. The exact scores calculated by MENDEL are plotted in figure 10. Also graphed in this figure are estimated location scores computed using the descent graph method. The difference between the simulated and exact results is always <0.1 and usually <0.04. Calculation of the simulated scores took a little under 2 h on a 40 MHz SuperSparc CPU. The deterministic calculation of the exact scores require >22 h on the same machine, because MENDEL's very large memory requirements forced the computer to continuously swap between RAM and storage. Exact scores cannot be computed within practical memory limitations if there are any additional loci or substantially more missing phenotypes. The descent graph method can, of course, handle much larger and sparser data sets.

To test the new location score method on a larger data set, we examined four of the 176 families in the A-T international consortium database (Lange et al. 1995). The four highly consanguineous pedigrees employed, ISAT 1, ISAT 3, ISAT 8, and ISAT 9 are diagrammed in the study by Ziv et al. (1992). These pedigrees range in size from 16 to 67 people and lack phenotyping on over one-third of their locus-person combinations. The results of our Markov chain simulation with 10 highly polymorphic markers are graphed as location scores in figure 11. In a previous study the entire consortium database—176 pedigrees and 18 markers—was analyzed...
using the descent state method, and the A-T gene was localized to an \( \approx 500 \) kb interval centered midway between the two markers D11S355 and D11S384, which are separated by \( \approx 200 \) kb (Lange et al. 1995). (Recall that 1,000 kb \( \approx 1 \) cM.) Recently, Savitsky et al. (1995) have cloned the A-T gene and found it scattered over an \( \approx 100 \)-kb-wide interval extending from just proximal to D11S355 toward D11S384. As seen in figure 11, the current test results on a small subset of the consortium database are consistent with these findings.

Finally, for testing the clustering statistics, we examined the pedigree data that enabled Hall et al. (1990) to map the BRCA1 breast cancer gene to 17q21. Hall et al. (1990) thoughtfully provide the pedigree structure and phenotypes for all their 23 pedigrees. These data exhibit genetic heterogeneity. The early-onset families 1–7, with an average age at onset \( \leq 45 \) years, support linkage. Many of the late-onset families contradict close linkage. Using the affected-pedigree-member (APM) statistic of Weeks and Lange (1988) on “the younger families,” presumably 1–7, Hall et al. (1990, p. 1689) found an empirical \( p \)-value of .001 with the reciprocal square root weight for matching

**Figure 8** Haplotyping of the EA pedigree under the CEPH ordering of 9 codominant chromosome 12p markers. Individuals 2001 and 1011 are completely untyped in the data set. Nonfounder haplotypes are ordered, maternal preceding paternal. The solid-black symbols indicate affecteds. The arrows indicate recombination sites. The boxed haplotypes show the conserved, disease-bearing chromosome segment.
alleles. Using all 23 families, Hall et al. cite a \( p \)-value of <.001. This second \( p \)-value may be too low because it assumes the normality of the test statistic.

Our clustering statistics are presented in Table 1. For the reason noted earlier, these empirical \( P \) values are conservative. It is noteworthy that in the face of considerable genetic heterogeneity our new clustering statistics exhibit even stronger evidence for linkage than the APM statistic. Undoubtedly, the additional information provided by the marker types of the unaffecteds prove crucial in this regard. The APM statistic ignores the marker types of the unaffecteds. Calculating our four statistics on all 23 pedigrees takes slightly under 2 h on a 125 MHz HP-PA CPU.

### Table 1

<table>
<thead>
<tr>
<th>Statistic for Families</th>
<th>1–7</th>
<th>1–23</th>
</tr>
</thead>
<tbody>
<tr>
<td>1–7</td>
<td>136.1450</td>
<td>.0411</td>
</tr>
<tr>
<td>2–3</td>
<td>140.5842</td>
<td>.0917</td>
</tr>
<tr>
<td>3–4</td>
<td>29.7186</td>
<td>.0001</td>
</tr>
<tr>
<td>4–5</td>
<td>93.1019</td>
<td>.0031</td>
</tr>
<tr>
<td>5–6</td>
<td>11.5215</td>
<td>.0026</td>
</tr>
<tr>
<td>6–7</td>
<td>37.1058</td>
<td>.0101</td>
</tr>
<tr>
<td>7–8</td>
<td>5.1065</td>
<td>.0002</td>
</tr>
<tr>
<td>8–9</td>
<td>15.7088</td>
<td>.0028</td>
</tr>
</tbody>
</table>
marker loci are highly polymorphic. Thus, it has been our experience that the two methods perform comparably on haplotyping and location score problems with complete typing data. On problems with substantial missing data, the descent graph method performs considerably better.

Unfortunately, the descent graph method entails more computational overhead. Although descent graph random walks take fewer steps to sample the search space adequately, each step takes more time. Our current descent graph program takes approximately three times longer per step than our more mature descent state program. However, the descent graph program typically gets much better results while using only half as many steps. Further algorithmic tweaking may reduce the disparity in speed per step.

Our descent graph program benefits from the incorporation of the new $T_2$ transition rules. This incorporation permits bigger steps and faster mixing of the Markov chain. One can imagine adding further composite transition rules. For example, a $T_1$ rule would exchange subtrees among the four grandparents of a given person. For the sake of program simplicity, we have only implemented the rules $T_0$, $T_1$, and $T_2$.

Although our clustering statistics can also be calculated by running a Markov chain over descent states, the statistics themselves are intrinsically measures of identity by descent among affecteds. In contrast to the APM statistic (Weeks and Lange 1988), the clustering statistics take advantage of the marker phenotypes among the unaffecteds of a pedigree. Possibly a compensating strength of the APM statistic is that it can detect marker-gene association as well as linkage.

There is room for improvement in our understanding of the distributions of the clustering statistics and of how to generalize them to multiple markers. For the present, we believe our suggested interpretation of empirical $P$ values is conservative and therefore serves a useful purpose in exploratory data analysis. Because the alternatives to independent segregation of marker genes and a disease phenotype are vague, it is doubtful that anyone will devise an optimal method for combining a given statistic across pedigrees.

Our list of clustering statistics is not meant to be exhaustive. For instance, it might be useful to design statistics reflecting the path lengths of the founder-tree graphs appearing among affecteds. As for statistics based on multiple markers, one obvious option is to sum a given statistic across markers. But this problem bears more careful thought and probably deserves a separate paper.

In closing, we note that the source code for the software described here, together with documentation and example usage, is available at the Internet site ftp://watson.hgen.pitt.edu/pub/simwalk2. The program MENDEL used in computing location scores can be obtained by contacting Kenneth Lange at the postal or electronic address listed in the footnote on the first page of this article. We hope readers will share our enthusiasm for these powerful computational tools inspired by the pioneering suggestion of Elizabeth Thompson.

**Acknowledgments**

This work was supported in part by funds from the University of Michigan and USPHS grant GM-53275 (K.L.). We thank Michael Litt and Richard Gatti for permission to use their EA and A-T pedigrees. We also thank Daniel Weeks and the Wellcome Trust Centre for Human Genetics for computing data and Michael Boehne and Eric Schadt for their suggested revisions of our preliminary manuscript.

**Appendix**

In this appendix we formalize the definition of the $T_2$ transition. We also demonstrate the symmetry of the proposal matrix $Q = (q_{ij})$ and thus the reversibility of the Markov chain on descent graphs.

A.1 Formalizing the $T_2$ Transition Rules

We execute a $T_2$ transition by choosing a locus $l$, a person $i$, and a spouse $j$ of $i$. We then swap the subtrees descending from $i$ at locus $l$ with those descending from $j$ at locus $l$. To effect this swap, we visit each common child $k$ of $i$ and $j$. In a $T_{2a}$ transition, we perform a $T_0$ transition at each node of $k$ at locus $l$ if and only if $k$'s two genes at this locus originate in $i$ and $j$ from nodes of opposite gender. In a $T_{2b}$ transition, we perform a $T_0$ transition at each node of $k$ at locus $l$ if and only if $k$'s two genes at this locus originate in $i$ and $j$ from nodes of the same gender. Once these $T_0$ transitions are either rejected or carried out, we perform a $T_1$ transition at locus $l$ of $k$. See figure 6 for a graphical illustration of how a $T_2$ transition works.

A.2 Symmetry of the Proposal Matrix

To prove the symmetry of the proposal matrix $Q = (q_{ij})$, it is convenient to interpret the set of multiple transitions as a transformation group acting on the descent graphs of a pedigree. Under a single-locus model, we designate a transition $T_{(r,c)}$ by its transition rule $r$ and choice $c$ of node, person, or couple. The index $r$ takes the values 0, 1, 2a, or 2b. If $r = 0$, then $c$ specifies a person and either his maternal or paternal node. If $r = 1$, then $c$ specifies a person, and if $r = 2$, then $c$ specifies a couple. The group product $T_{(r_2,c_2)} \circ T_{(r_1,c_1)}$ corresponds to the step that performs transition $T_{(r_1,c_1)}$ first and then transition $T_{(r_2,c_2)}$. In general, the product

$$T_{(r_3,c_3)} \circ \ldots \circ T_{(r_2,c_2)} \circ T_{(r_1,c_1)}$$

(7)
corresponds to the step that performs \( T_{(r_1,c_1)} \) first, then \( T_{(r_2,c_2)} \), and so forth, until finally \( T_{(r_k,c_k)} \). In standard algebraic language, the transitions \( T_{(r,c)} \) generate the transformation group. Because each transition undoes itself, as is evident in figures 4–6, the inverse of the step (7) is the step

\[
T_{(r_1,c_1)} \circ T_{(r_2,c_2)} \circ \ldots \circ T_{(r_k,c_k)} .
\] (8)

Suppose that the number of transitions \( k \) of a step is chosen with probability \( q_k \) and that a particular transition \( T_{(r,c)} \) is chosen for a step with probability \( \alpha(r, c) \). If each transition is chosen independently, then step (7) is selected with probability \( q_k \alpha(r_1, c_1) \ldots \alpha(r_k, c_k) \). The inverse step (8) is chosen with equivalent probability \( q_k \alpha(r_1, c_1) \ldots \alpha(r_k, c_k) \). Thus, to each step from descent graph \( G_i \), to descent graph \( G_t \), there corresponds an inverse step with the same probability. Summing these probabilities over all possible steps between the two descent graphs shows that \( q_{ij} = q_{ji} \).

For a multilocus model, we must add a locus subscript \( l \) to a transition \( T_{(r,c,l)} \). The set of steps can still be viewed as a transformation group. Furthermore, we have the additional commutativity property

\[
T_{(r_2,c_2,l_2)} \circ T_{(r_1,c_1,l_1)} = T_{(r_1,c_1,l_1)} \circ T_{(r_2,c_2,l_2)}
\] (9)

when \( l_1 \neq l_2 \). We assume that the locus \( l \) of a transition \( T_{(r,c,l)} \) is selected independently of the combination of rule \( r \) and node, person, or couple choice \( c \). Correlated choices within the sequence of loci of a step

\[
T_{(r_2,c_2,l_2)} \circ \ldots \circ T_{(r_k,c_k,l_k)} \circ T_{(r_1,c_1,l_1)}.
\] (10)

are possible, but not correlated choices within the sequence of \( (r, c) \) combinations. Let \( \beta(l_1, \ldots, l_k) \) be the probability of selecting the sequence of loci \( l_1, \ldots, l_k \). Then capitalizing on our notation in the single-locus situation, we find that step (10) has probability \( q_k \beta(l_1, \ldots, l_k) \alpha(r_1, c_1) \ldots \alpha(r_k, c_k) \).

In order to prove symmetry of the \( Q \) matrix in this multilocus setting, it suffices to exhibit an inverse to step (10) with the same probability. In rather opaque notation, the inverse step is

\[
T_{(r_1,c_1,l_1)} \circ \ldots \circ T_{(r_k,c_k,l_k)} \circ T_{(r_1,c_1,l_1)} .
\] (11)

The important thing to note about this step is that we take the loci in the same order, but scramble the \( (r, c) \) choices. In fact, we make the \( (r, c) \) choices within a given locus in the reverse order of the choices appearing in step (10). To avoid introducing cumbersome notation to explain the idea, let us consider a simple example with two loci and five transitions. If the step, corresponding to (10), is

\[
T_{(r_1,c_1,2)} \circ T_{(r_2,c_2,2)} \circ T_{(r_3,c_3,1)} \circ T_{(r_4,c_4,2)} \circ T_{(r_5,c_5,1)} ,
\]

then the inverse step, corresponding to (11), is

\[
T_{(r_5,c_5,1)} \circ T_{(r_4,c_4,2)} \circ T_{(r_3,c_3,1)} \circ T_{(r_2,c_2,2)} \circ T_{(r_1,c_1,2)} .
\]

As this example illustrates, the two steps (10) and (11) have the same probability. Step (11) is the inverse of step (10) because of the commutativity property (9), which permits us to regroup transitions by locus, and because within a locus a sequence of transitions is undone by the same sequence taken in reverse order. Symmetry of the \( Q \) matrix now follows just as for a single locus.

References


Sobel and Lange: Descent Graphs in Pedigree Analysis

for multilocus linkage analysis in humans. Proc Natl Acad Sci USA 81:3443–3446