Statistics and Computing (2000) **10,** 259–269

Multilocus linkage analysis by blocked Gibbs sampling

ALUN THOMAS[*,†], ALEXANDER GUTIN[*], VICTOR ABKEVICH[*]
and ARUNA BANSAL[†,**]

[*]*Myriad Genetics Inc., 320 Wakara Way, Salt Lake City, UT 84108, USA*
[†]*Genetic Epidemiology, Department of Medical Informatics, University of UT, 391 Chipeta Way D2, UT 84108, USA*
[**]*IHC Genetic Research, 391 Chipeta Way C, UT 84108, USA*

Received October 1998 and accepted August 1999

The problem of multilocus linkage analysis is expressed as a graphical model, making explicit a previously implicit connection, and recent developments in the field are described in this context. A novel application of blocked Gibbs sampling for Bayesian networks is developed to generate inheritance matrices from an irreducible Markov chain. This is used as the basis for reconstruction of historical meiotic states and approximate calculation of the likelihood function for the location of an unmapped genetic trait. We believe this to be the only approach that currently makes fully informative multilocus linkage analysis possible on large extended pedigrees.

Keywords: genetic linkage, graphical models, Markov chain Monte Carlo, peeling

1. Introduction

Human gene mapping is based on observing correlations between disease segregation and transmission of alleles at genetic markers. Allele transmissions, however, are not directly observable and have to be inferred from the genotypes of individuals involved. In animal genetics, where the matings can be appropriately arranged, the transmissions can be inferred completely, but in human studies inferences are incomplete and leave a large possible state space. Analysis of several proximal genetic markers in large extended families can yield powerful statistical methods to address the problem and is essential in inferring the positions of meiotic recombination events and hence in fine scale gene localization.

The problem can be referred to as the *haplotyping problem*, in which the unordered pair of alleles that make up a person's observable genotypes for a set of close markers are assigned to the paternally or maternally inherited chromosome. This is also known as determining the *phase* of the alleles at each locus. The information at each locus can be summarised as an *inheritance vector* (Lander and Green 1987), also called a *genetic descent graph* (Sobel and Lange 1996), in which a vector of binary variables indicates for each transmission whether the parent's maternal or paternal allele was inherited by the child. The collection of inheritance vectors, one for each locus, is, in the formal statistical sense, sufficient for the problem. We shall call this the *inheritance matrix*.

The genetic distance between two genetic loci is usually parameterised by the probability that a genetic recombination event occurs between them. For the problem we consider, we assume that the markers we analyse are at known positions, and that only the position of the disease, parameterised by the probability θ, has to be estimated. The likelihood framework for this problem was developed by Morton (1955) where the likelihood function, $L(\theta)$ is usually reported in the form of a *Lod* function, defined as

$$\mathrm{Lod}(\theta) = \log_{10} \frac{L(\theta)}{L\left(\frac{1}{2}\right)} \tag{1}$$

Frequently, only the *Lod score* or maximum of this function over θ is reported.

In principle, the likelihood function could be calculated by summing the contribution from each inheritance matrix, weighted by its probability given the observed marker phenotypes. Letting D be the vector of observed phenotypes for the trait being linked, I the inheritance vector for the hypothesised underlying trait locus, M the matrix of observed marker data, H the inheritance matrix for the markers, and ϕ the vector of

known recombination fractions between the marker loci,

$$L(\theta) = P(D, M \mid \theta) \propto \sum_{\text{all } H} \sum_{\text{all } I} P(DI \mid H\theta) P(H \mid M\phi) \qquad (2)$$

The state space of all inheritance vectors grows exponentially with the number of individuals in the family and with the number of genetic loci considered. Despite this, methods have been developed to implement calculation (2) in a surprisingly large number of cases by exploiting conditional independence. Elston and Stewart (1971) introduced a method which allowed such calculations to be made in unlooped pedigrees by successively collapsing information from offspring into probability distributions on parents. This uses the conditional independence of a child's genotype and a grandparent's genotype given the genotype of the intervening parent. The method was extended to other pedigree structures by Lange and Elston (1975) and to fully general pedigrees by Cannings, Thompson and Skolnick (1978). Cannings *et al.* (1978) called the method *peeling*. While the peeling method is now entirely general, computational requirements grow as an exponent of the complexity of a pedigree, or how looped it is. Programs such as LINKAGE (Lathrop *et al.* 1984) and FASTLINK (Cottingham, Idury and Schaffer 1993) are based on the Lange and Elston (1975) approach, and most applications involve unlooped pedigrees. For these cases computations grow linearly in the number of people in the pedigree, but exponentially in the number of loci considered. In practice, using the VITESSE program (O'Connell and Weeks 1995), simultaneous consideration of about 5 loci is the limit of current feasibility.

A far more recent, and orthogonal, development has been that of Lander and Green (1987) implemented by the GENEHUNTER program (Kruglyak *et al.* 1996). This exploits the common assumption that recombination events occur as a Poisson process along a chromosome during meiosis, usually termed *no interference* in genetics. The result of this is that the inheritance vector at a particular locus is independent of those at other loci, conditional on the vectors for the two adjacent loci. In other words, the columns of the inheritance matrix have a first order Markov property, provided the ordering of the columns corresponds to the physical order of the marker loci. This allows the required computation to grow linearly with the number of loci, but exponentially in the number of people in the pedigree. In the current implementation pedigrees of more than around 25 people are infeasible.

This problem can be expressed as a graphical model (Lauritzen and Spiegelhalter 1988) and it is clear that likelihood calculation by both peeling and the Lander and Green methods are special cases of the information collecting and dispersing phases of this methodology. The approaches differ in the variables modelled in the graph, and the triangulation of the graph implicit in the order in which variables are summed out. Both methods use distributional symmetries to speed up calculation and GENEHUNTER also makes innovative use of Fourier transforms for efficiency gains (Kruglyak and Lander 1998). Despite this, calculations are still intractable for more than about 5 markers in pedigrees of more than about 25 people.

Various Markov chain Monte Carlo methods have been used to address particular genetic problems (Sheehan 1990, Sheehan and Thomas 1993, Sheehan 1992, Sobel and Lange 1996) and more general graphical models (Gelfand and Smith 1990, Thomas, Spiegelhalter and Gilks 1992, Gelman and Rubin 1992, Geyer 1992, Smith and Roberts 1993). A recent development has been that of efficient blocked Gibbs sampling for graphical models (Jensen, Kong, and Kjaerulff 1995, Jensen and Kong 1996, Jensen 1997) which allows updating of large subgraphs of the model simultaneously. We express the multilocus linkage problem explicitly as a graphical model and apply blocked Gibbs sampling as an efficient method of reconstructing inheritance matrices and approximate calculation of Lod functions for arbitrarily large pedigrees and an arbitrarily large number of loci. Blocking not only updates a large number of variables simultaneously but also, in the way we apply it, ensures irreducibility of the induced Markov chain.

In Section 2 we review the aspects of graphical model methods relevant to our problem including blocked Gibbs sampling. We formulate the linkage problem as a graphical model, describe the Markov chain Monte Carlo updates made, and specify how the inheritance matrices generated are used for likelihood calculations. Subsection 2.3 also serves as a review of the development of likelihood calculation for linkage genetics in terms of the underlying graphical models. Section 3 illustrates the method in an analysis of data from several chromosome 14 markers, comparing our results with those from other methods. Finally, in Section 4, we summarise our findings and describe areas of future work.

Although the genetic problem and peeling predate the general graphical model approach, we shall use the terminology of the more general method indicating, where appropriate, corresponding terms or methods used in genetics.

2. Materials and methods

2.1. *Graphical models and efficient computation*

Lauritzen and Spiegelhalter (1988) describe a methodology for efficient propagation of information on Gaussian random variables and discrete random variables with finite range when the joint probability of any state $\underline{x} = (x_1, x_2, \ldots, x_n)$ can be factorised as

$$P(\underline{x}) = \prod_{i=1}^{n} P(x_i \mid C(x_i)). \qquad (3)$$

$C(x_i)$ is a relatively small set of variables, called the *parent* variables of x_i . The directed graph which joins parent variables to offspring variables must be acyclic. Information is input in the

form of constraints on some or all of the variables. For computational purposes, however, the relevant graph is the so called *moral graph*, an undirected graph in which offspring variables are joined to their parents, and parents of the same offspring are joined to each other. Efficient computations are determined by triangulating the moral graph, finding the cliques and clique intersections of the triangulated graph, and deriving a junction tree (Jensen 1988). The junction tree is a maximal spanning tree of the graph that has cliques as nodes which share an edge if they intersect, the weight of the edge being the size of the intersection. The most efficient order in which to make computations of joint probability distributions on subsets of variables can then be read from the junction tree. While Lauritzen and Spiegelhalter (1988) formulate the problem in a Bayesian framework, it is equally applicable for computation of likelihoods in other contexts. Further reading about graphical models and Bayesian networks can be found in Jensen (1996) and Lauritzen (1996).

Several operations can now be performed efficiently. The first step is to *collect evidence*. This operation collects all the information in the graph towards a single clique, successively summing out the variables in an order derived from the junction tree. Summing over all the terms of the distribution on the final clique gives the probability of the subspace determined by the input constraints. When the probabilities depend on the value of a parameter the result is the likelihood for that parameter given the observed data, and this operation is exactly equivalent to the peeling method as developed by Cannings *et al.* (1978). As this operation is performed, the intermediate conditional probability functions on the cliques and clique intersections are stored to prepare for subsequent operations.

The next step is usually to *distribute evidence*. This operation reverses the order of the collect evidence step and successively calculates the marginal distribution for each clique and clique intersection, given the input constraints. This method is not usually necessary for likelihood computations which can be made in a single pass.

The distribute evidence step may be replaced by *random propagation* (Dawid 1992) where instead of calculating marginal distributions, we successively generate random realizations from them. This results in a random realization from the distribution of states given the input constraints. An analogous method was used by Ploughman and Boehnke (1989) in assessing the power of pedigrees for linkage analysis.

It is also possible, as described by Dawid (1992), to replace summation over states with maximisation and hence find the most likely configuration in the graph, essentially generalised dynamic programming.

The computational resources required for all of these operations are determined by the size of clique state spaces, and are dominated by the largest clique state space. When peeling a single locus trait on an unlooped pedigree, the cliques are formed by the parent/offspring genotype triplets. Hence, the time required is of order ng^3 where n is the number of individuals in the pedigree and g is the number of genotypes for the trait.

2.2. *Blocked Gibbs sampling*

Markov chain Monte Carlo sampling methods, and particularly the Gibbs sampler (Geman and Geman 1984), have been used extensively for graphical models in many areas of interest including genetics. Sheehan (1990) first developed the Gibbs sampler for the case of single biallelic loci, Sheehan and Thomas (1993) developed irreducible schemes for sampling from arbitrary genetic loci and Thomas (1994) applied this to the two locus linkage problem.

In a graphical model with n variables $N = \{1, 2, \ldots, n\}$, the Gibbs sampler proceeds by updating the state of each variable in turn by simulating from $p(x_i \mid x_{-i})$, the conditional distribution of the ith variable given the current state of all other variables. Jensen (1997) extended this to simulating efficiently from $p(x_I \mid x_{\bar{I}})$ the conditional distribution of a set of variables indexed by $I \subseteq N$ given the current state of all other variables, the set $\bar{I}$, in complex graphical models. Updates are made by considering only the subgraph of the moral graph induced by the variables I, the rest of the graph being unnecessary since its state is fixed by conditioning. Collect information and random propagation steps are then performed on the induced subgraph. By choosing I such that the induced subgraph is loosely connected, a simulation from a state space whose size is exponential in $|I|$ can be made using time and storage requirements which are linear, or approaching linear, in $|I|$. Moreover, it should be possible to choose a sequence of update sets each of which contains a large proportion of the graph. There typically will be a trade off between the proportion of the graph included in any update and the time and storage required to make the update. Of course, the union of the chosen sets must be N.

2.3. *The linkage problem as a graphical model*

Figure 1 shows an example of a simple nuclear family with two children in the usual format. For a single locus problem this would usually be parameterised in terms of genotypes, the unordered pair of alleles present for an individual at the locus in question, and phenotypes, the observable trait dependent on the genotype. For most genetic markers there is a one to one correspondence between genotype and phenotype, that is they are co-dominant, but this is not necessary for our method. Assuming independent segregation of genotypes from parents to offspring, and that the observed phenotypes depend only on the underlying genotype, not on any shared environmental factors, the probability of any state for the variables can be factorised as

$$\begin{aligned} P(\underline{M}, \underline{G}) &= P(M_1 .. M_n, G_1 .. G_n) \\ &= \prod_{i \in F} P(G_i) \prod_{j \in \bar{F}} P\left(G_j \mid G_{f_j}, G_{m_j}\right) \prod_{k \in N} P(M_k \mid G_k) \end{aligned} \tag{4}$$

where N is the set of indices $\{1 .. n\}$ corresponding to the n individuals in the pedigree, F is the set of indices for the f founders

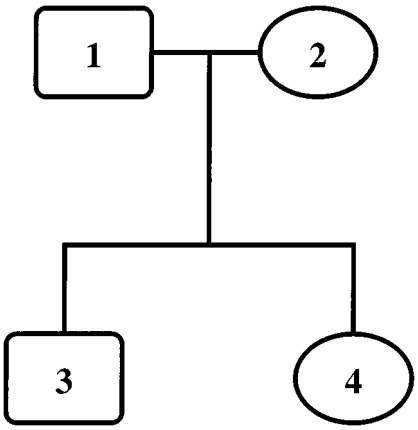

Fig. 1. *A nuclear family with two children*

of the pedigree, G_i is the genotype of the ith individual, f_j and m_j are the parents of the jth individual, and M_k represents the observed marker data, or phenotype, for the kth individual. $P(G_i)$ typically would be genotype frequencies based on Hardy-Weinberg equilibrium, and $P(G_j \mid G_{f_j}, G_{m_j})$ would encode Mendelian inheritance probabilities. $P(M_k \mid G_k)$ is usually a one to one function, but it might also reflect mistyping error probabilities, or partial observations and is usually referred to as the *penetrance* function. In many cases the markers for many people will not have been observed, making this function uniform on all genotypes. It is then simply omitted in efficient implementations. Figure 2 illustrates the corresponding moral graph for the example family.

The above parameterisation does not express all available conditional independences, in particular, the transmission of an allele from one parent to an offspring is independent of the transmission from the other parent. Harbron and Thomas (1994) examined alternative parameterisations in terms of the ordered alleles for individuals at a locus that express this additional independence. This parameterisation was previously used by (Kong 1991). Letting A_{2i-1} and A_{2i} respectively be the paternally and maternally inherited alleles for the ith individual we then get the following factorisation

$$\begin{aligned} P(\underline{M}, \underline{A}) = & \prod_{i \in F} P(A_{2i-1}) P(A_{2i}) \\ & \times \prod_{j \in \bar{F}} P\left(A_{2j-1} \mid A_{2f_{2j-1}-1}, A_{2f_{2j-1}}\right) \\ & \times P\left(A_{2j} \mid A_{2m_{2j}-1}, A_{2m_{2j}}\right) \\ & \times \prod_{k \in N} P(M_k \mid A_{2k-1}, A_{2k}) \end{aligned} \tag{5}$$

or more simply

$$\begin{aligned} P(\underline{M}, \underline{A}) = & \prod_{i \in E} P(A_i) \prod_{j \in \bar{E}} P\left(A_j \mid A_{f_j}, A_{m_j}\right) \\ & \times \prod_{k \in N} P(M_k \mid A_{2k-1}, A_{2k}) \end{aligned} \tag{6}$$

where E is the set of $e = 2f$ founding alleles of the founders F. We now use f_j and m_j to index the parental alleles from which the jth allele is drawn. $P(A)$ is the frequency of the allele A in the population that the founders are drawn from and

$$P\left(A_j \mid A_{f_j}, A_{m_j}\right) = \begin{cases} 1 & \text{if } A_j = A_{f_j} = A_{m_j} \\ \dfrac{1}{2} & \text{if } A_j = A_{f_i} \text{ or } A_j = A_{m_j},\ A_{f_j} \neq A_{m_j} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

If $P(M_k \mid A_{2k-1}, A_{2k}) = P(M_k \mid A_{2k}, A_{2k-1})$ as is the case unless genetic imprinting is assumed in the model, then the two parameterisations (4) and (6) are equivalent. The moral graph for this is given in Fig. 3. Using this parameterisation, Harbron and Thomas (1994) showed that all the steps required for the graphical model operations needed to deal with a single locus with a alleles on an unlooped pedigree of n individuals can be done in time proportional to $na^4(a+1)$. For $a \geq 6$ this compares favourably with the genotype parameterisation which takes time proportional to $n(\frac{a(a+1)}{2})^3$.

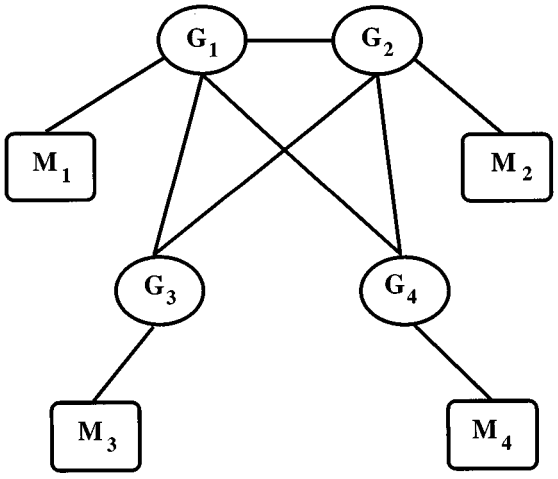

Fig. 2. *The moral graph for a single locus parameterised with genotypes*

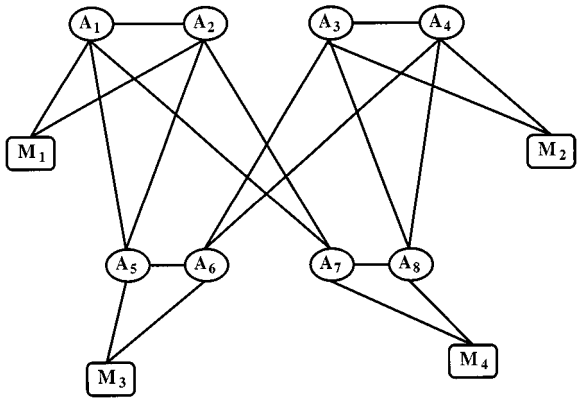

Fig. 3. *The moral graph for a single locus parameterised with alleles*

In order to generalise the allelic model for several loci we follow Jensen (1997) first adding an indicator variable H_j for each meiosis to get

$$P(\underline{M}, \underline{A}, \underline{H}) = P(\underline{M}, \underline{A} \mid \underline{H}) \prod_{j \in \bar{E}} P(H_j) \tag{8}$$

where

$$P(\underline{M}, \underline{A} \mid \underline{H}) = \prod_{i \in E} P(A_i) \prod_{j \in \bar{E}} P\left(A_j \mid A_{f_j}, A_{m_j}, H_j\right) \times \prod_{k \in N} P(M_k \mid A_{2k-1}, A_{2k}) \tag{9}$$

$$P(H_j) = \begin{cases} \frac{1}{2} & \text{if } H_j \in \{0, 1\} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

and

$$P\left(A_j \mid A_{f_j}, A_{m_j}, H_j\right) = \begin{cases} 1 & \text{if } A_j = A_{f_j} \text{ and } H_j = 1 \\ 1 & \text{if } A_j = A_{m_j} \text{ and } H_j = 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The moral graph for this case is given in Fig. 4, from which it is clear that summing over the $\{H_i\}$ returns us to the previous allelic parameterisation (6). Moreover, it also shows that the additional computational burden due to the extra variables is of order $2na^3$, which is negligible compared with the $o(na^5)$ time required to deal with an unlooped pedigree.

In order to describe the multilocus case we extend our notation so that the matrix $A = \{A_{i,j}\}$, where $A_{i,j}$, is the ith allelic variable for the jth locus, and $\underline{A}_j$ is the jth column of A, with similar interpretations for M and H. Then, under the usual assumption that recombinations occur as a Poisson process along the chromosome, we get the following.

$$P(M, A, H) = \prod_{j=1}^{m} P(\underline{M}_j, \underline{A}_j \mid \underline{H}_j) P(\underline{H}_1) \prod_{k=2}^{m} P(\underline{H}_k \mid \underline{H}_{k-1}) \tag{12}$$

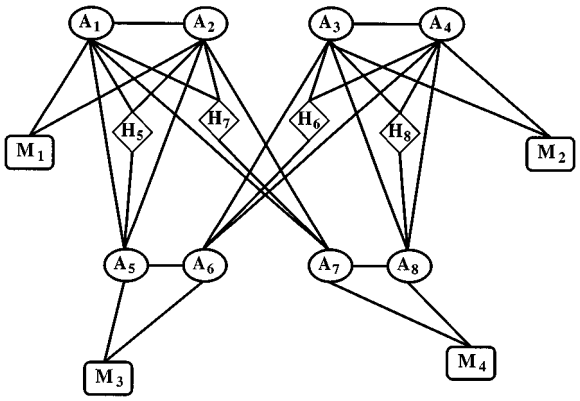

Fig. 4. *The moral graph for a single locus parameterised with alleles and meiotic states*

where $P(\underline{M}_j, \underline{A}_j \mid \underline{H}_j)$ is given by equation (9),

$$P(\underline{H}_1) = \frac{1}{2^{2(n-f)}} \text{ for all values of } \underline{H}_1, \tag{13}$$

$$P(\underline{H}_k \mid \underline{H}_{k-1}) = \prod_{i \in \bar{E}} P(H_{i,k} \mid H_{i,k-1}), \tag{14}$$

and

$$P(H_{i,k} \mid H_{i,k-1}) = \begin{cases} \phi_{k-1,k} & \text{if } H_{i,k} \neq H_{i,k-1} \\ 1 - \phi_{k-1,k} & \text{if } H_{i,k} = H_{i,k-1} \\ & \text{for } 0 < \phi_{k-1,k} \leq \frac{1}{2} \end{cases} \tag{15}$$

Figure 5 gives the corresponding moral graph. This is the structure implicit in Sobel and Lange (1996) and Lander and Green (1987). Sobel and Lange (1996) define a set of transition rules on the graphical model in which subsets of the $\underline{H}_j$ are changed using a sequence of smaller changes which may pass through states of zero probability. For the Lander and Green (1987) method,

$$P(\underline{M}_j \mid \underline{H}_j) = \sum_{\text{all } \underline{A}_j} P(\underline{M}_j, \underline{A}_j \mid \underline{H}_j) \tag{16}$$

is calculated and stored for each locus. These values are then combined using the locus to locus transition probabilities $P(\underline{H}_k \mid \underline{H}_{k-1})$, Fourier transforms can be used to make the transition calculations efficient (Kruglyak and Lander 1998). By exploiting symmetries in the state space of $\underline{H}_j$, the storage and time requirements to calculate the $P(\underline{M}_j \mid \underline{H}_j)$ grow as $2^{2(n-f)-f}$ rather than as $2^{2(n-f)}$. However, this still limits application of the method to pedigrees of around 25 individuals. Viewed as operations on vectors of variables, the moral graph corresponding to this is given by Fig. 6. The original application of this method was for the situation when the order of the loci was known but the genetic distances between them not. In that case, iterations of the EM-algorithm (Dempster, Laird, and Rubin 1977) were used to derive maximum likelihood distance estimates.

2.4. *The block updating scheme*

Conditional on the inheritance vectors $\underline{H}_{j-1}$ and $\underline{H}_{j+1}$, and the observed marker data $\underline{M}_j$, a collect information and random propagation step for $\underline{H}_j$ and $\underline{A}_j$ reduces to a simple single locus operation. In genetic terms we peel the locus and hence generate a new realisation. For an unlooped pedigree this operation can be performed in time linear in the size of the pedigree. The same method can, of course, be used on pedigrees with an arbitrary number of loops but computational requirements will grow exponentially with the complexity of the pedigree as measured by the size of the largest clique in the triangulated moral graph.

This is the basis of our block updating scheme, the blocks consist of the allele and inheritance variables relevant to each locus. In essence we treat the columns of H as single variables, but in such a way that the exponentially large state space for each column variable is sampled in linear time. The corresponding moral

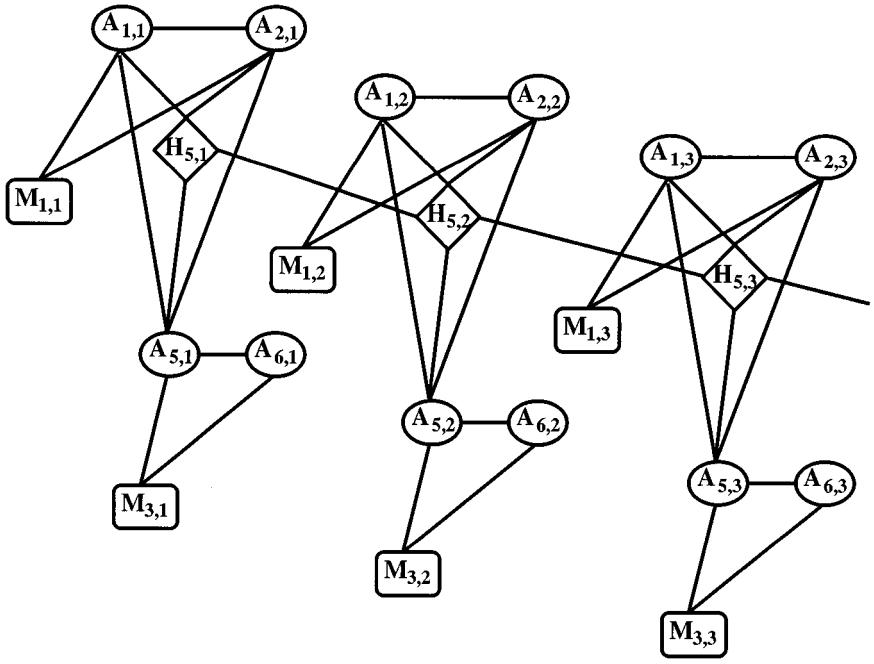

Fig. 5. *Part of the moral graph for the multilocus problem parameterised with alleles and meiotic states*

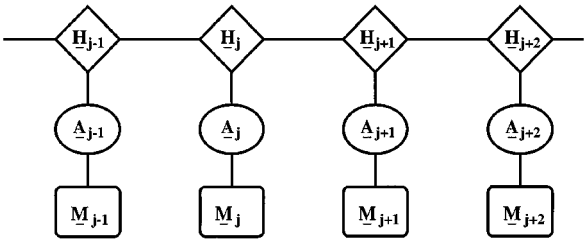

Fig. 6. *Part of the moral graph for the multilocus problem parameterised as vectors*

graph is given in Fig. 6. This vectorisation of the inheritance matrix by column is implicit in the Lander and Green (1987) method, but since the probability for every state is stored, the storage requirement is exponential in the column length. Peeling conversely, as it is applied to linkage analysis, can be viewed as a vectorisation of H by rows.

From equations (12) to (15) it is clear that the state space for the inheritance matrix given all marker data is the product of the state spaces for the inheritance vectors at each locus given only the marker data at the corresponding locus. That is, $P(M, A, H) > 0$ if and only if $P(\underline{M}_j, \underline{A}_j, \underline{H}_j) > 0$ for all j. Thus, our block updating scheme which visits all loci, updating all elements of the vectors $\underline{A}_j$ and $\underline{H}_j$ simultaneously, must define an irreducible Markov Chain and so has the appropriate ergodic distribution. This guarantee of theoretical irreducibility makes our method fundamentally different to existing methods which change individual variables, or small blocks of variables, in each update.

A starting configuration for our scheme can be generated in several ways. By the above observation, simply performing an unconditional collect evidence and random propagation at each locus, using only information at that locus, will give us a multilocus configuration with positive probability. Alternatively, we can perform an unconditional collect evidence and random propagation step on some locus, then, incorporating loci in an arbitrary order, make a similar step for each locus conditioning only on loci for which states have already been generated, essentially, using the method of sequential imputation (Kong *et al.* 1993) as a starting point for Monte Carlo Markov chain simulation. Either way we can generate a starting configuration using the same program steps as the main simulation scheme and avoid special treatment of this case by methods such as those of Heath (1998), Jensen (1998) or Sheehan and Thomas (1993).

2.5. *Additional block updates*

While our block updating scheme ensures theoretical ergodicity, in practice certain transitions may have probability too small to occur even in a long simulation run. Therefore, in order to have better mixing properties, we have introduced extra transitions which update the inheritance states of all loci simultaneously for a small set of meioses. That is, simultaneous updates of rows of H.

At any particular locus, as Lander and Green (1987) show, we can find the posterior distribution of $2(n - f)$ inheritance variables in time proportional to $2^{2(n-f)}$. Extending this idea we can also find the posterior distribution of a subset of k inheritance variables, conditional on particular values for the remaining $2(n - f) - k$ in time proportional to 2^k. This can be done by a collect evidence step in which the k variables of interest are not summed out but, once encountered in the calculations, kept

in as arguments to all subsequent functions including the output function. This is similar to the use of peeling to obtain likelihoods for ancestral genotypes. Combining information from locus to locus for this subset of k meioses can be done in time proportional to $k2^{k+1}$ using the recurrence relation

$$\begin{aligned} &P(\underline{M}_j \,..\, \underline{M}_m \mid \underline{H}_j) \\ &\quad = P(\underline{M}_j \mid \underline{H}_j) \sum_{\underline{H}_{j+1}} P(\underline{H}_{j+1} \mid \underline{H}_j) P(\underline{M}_{j+1} \,..\, \underline{M}_m \mid \underline{H}_{j+1}) \\ &\quad = P(\underline{M}_j \mid \underline{H}_j) \sum_{H_{1,j+1}} \cdots \sum_{H_{k,j+1}} \prod_{i=1}^{k} P(\underline{H}_{i,j+1} \mid \underline{H}_{i,j}) \\ &\qquad \times P(\underline{M}_{j+1} \,..\, \underline{M}_m \mid \underline{H}_{j+1}) \\ &\quad = P(\underline{M}_j \mid \underline{H}_j) \sum_{H_{1,j+1}} P(\underline{H}_{1,j+1} \mid \underline{H}_{1,j}) \\ &\qquad \times \sum_{H_{2,j+1}} P(\underline{H}_{2,j+1} \mid \underline{H}_{2,j}) \cdots \sum_{H_{k,j+1}} P(\underline{H}_{k,j+1} \mid \underline{H}_{k,j}) \\ &\qquad \times P(\underline{M}_{j+1} \,..\, \underline{M}_m \mid \underline{H}_{j+1}) \end{aligned} \tag{17}$$

In this way we perform a collect evidence step on a block containing all the variables in the allele matrix A and a subset of k rows of the inheritance matrix H conditioning only on the values of the remaining $2(n-f)-k$ rows of H, in time proportional to $mk2^{k+1}$ requiring storage of $mk2^k$ terms. A random propagation can be performed in similar time for an efficient blocked Gibbs update.

As a further refinement which allows more variables to be updated at each step we impose constraints to limit the number of new states possible. So, if changes on a group of s inheritance variables are controlled by a smaller number, t say, of binary variables we have 2^t possibilities at each locus rather than 2^s.

Note that in both column and row types of updates we condition only on subsets of H, and not on any of the allelic states A, which are always updated. To use the terminology and distinction of Sobel and Lange (1996), this is a method which updates *genetic descent graphs* not *genetic descent states*. This allows us freedom to traverse the state space of H unconstrained by particular simulated values of A.

The specific multilocus blocks that we update are defined by the following.

2.5.1. *Individuals*

Take in turn each individual who is not a founder. Update their maternal and paternal inheritances jointly. This has 2^2 possibilities at each locus.

2.5.2. *Nuclear families*

Take in turn each nuclear family in the pedigree. Define two binary variables at each locus, X_j and Y_j say. If X_j is true the inheritances at the jth locus from the father to each of the children are flipped. If Y_j is true the inheritances at the jth locus from the mother to each of the children are flipped. This update is also controlled by 2^2 possibilities at each locus. It has the effect of swapping the father's maternally and paternally inherited alleles and/or the mother's maternally and paternally inherited alleles in a joint step.

2.5.3. *Three generation families*

Take in turn each three generational family of grandparents, their children and their grandchildren. Define a single binary variable, X_j at each locus. If X_j is true the inheritances from all of the children to all of the grandchildren at the jth locus are flipped, and the maternal and paternal inheritances of the children are swapped. This update is controlled by only 2 possibilities at each locus. It has the effect of swapping the alleles of the grandfather and grandmother.

For each of the last two types of update most individuals will appear in multiple overlapping family blocks. Although these are very restrictive, and hence efficient, updates they have proved important in making a chain with good mixing properties.

2.6. *Implementational considerations*

Our implementation updates loci in random order, subject to the constraint that no locus is updated twice in immediate succession. After a number of updates equal to the number of loci a realisation is output, although not every block will necessarily have been updated since the last realisation used. Other schemes are obviously possible, such as updating the loci in some fixed order, or generating a random permutation of loci for each round of updates, but we would not expect the behaviour to differ appreciably.

The observation that the state space for the inheritance matrix is the product of the column state spaces allows us to significantly reduce the computational burden. A collect evidence and distribute evidence step performed on each locus in turn, using only the observations at the appropriate locus, can be performed at start up. This will allow us to identify the set of allelic states possible for each individual at each locus. Thus, by restricting future computations to only those states, we can reduce the computational requirement for an iteration on the jth locus from

$$na_j^4(a_j+1) \tag{18}$$

where a_j is the number of alleles at the jth locus to

$$\sum_{i \in N} a_{i,j}^4 (a_{i,j}+1) \tag{19}$$

where $a_{i,j}$ is the number of states possible for the ith allele at the jth locus. In a pedigree with dense genotyping information, this can increase speed considerably.

In fact, further refinements are possible. Conditional on the alleles of a parent, the allele that the child inherits can take only one of the two parental values. So by arranging the order of summation appropriately, we can replace a factor of $a_{i,j}$ in equation (19) by a factor of 2 for many steps.

It is also possible to find the feasible state space for a set of variables jointly, and when this is smaller than the product of the

individual state spaces it may be better to store these states and step through them in subsequent updates. This is usually true for the four alleles for each pair of mated individuals, and so we implement the saving in this case.

For unlooped pedigrees, the triangulation of the moral graph needed to determine the order of summation for the graphical model operations can be generated following the rules developed by Harbron and Thomas (1994). For looped pedigrees, some search and optimisation scheme may be necessary. Greedy algorithm methods have been shown to give adequate solutions but simulated annealing may do better and could be tried on more complex pedigrees (Thomas 1986). In any case, finding optimal triangulations needs to be done only once, the same solution then being used for all updates. If the number of alleles differ greatly it might be best to use different triangulations at each locus, but if they are about the same using the same suboptimal but good solution at all the loci will probably suffice.

All our calculations for computational requirements represent the worst case. In cases when information is available to exactly determine the value of a variable, that variable needs no further updating and any distributions conditional on it can be calculated at start up. This is expressed graphically by deleting the corresponding node and all adjacent edges from the moral graph, a structural change that can help produce more efficient triangulations for some loci.

2.7. *From inheritance matrix to Lod scores*

Given a sequence of s inheritance matrices $H^1 \,..\, H^s$ simulated from $P(H \mid M\phi)$, we can make an approximate calculation of (2) as

$$L(\theta) = \sum_{\text{all } H} \sum_{\text{all } I} P(D, I \mid H, \theta) P(H \mid M, \phi)$$

$$\approx \frac{1}{s} \sum_{t=1}^{s} \sum_{\text{all } I} P(D, I \mid H^t, \theta) \tag{20}$$

We calculate the inner sum exactly for each H^t and for a range of values of θ containing the whole genetic region spanned by the markers, summing over each possible inheritance state at the trait locus rather than simulating a subset of them. More explicitly, letting B be the vector of alleles at the trait locus, we have that

$$\sum_{\text{all } I} P(D, I \mid H^t, \theta)$$

$$= \sum_{\text{all } I} \sum_{\text{all } B} P(D \mid B) P(B \mid I) P(I \mid H^t, \theta) \tag{21}$$

$$= \sum_{\text{all } I} \sum_{\text{all } B} \prod_{i \in E} P(B_i) \prod_{j \in \bar{E}} P\left(B_j \mid B_{f_j}, B_{m_j}, I_j\right)$$

$$\times \prod_{k \in N} P(D_k \mid B_{2k-1}, B_{2k}) \prod_{l \in \bar{E}} P(I_l \mid H^t, \theta) \tag{22}$$

where the first three products are analogous to those in equation (9). Under the Poisson assumption of no interference, the inheritance at the disease locus for the lth segregation depends on H^t only through the inheritances for that segregation at the two loci adjacent to the putative position defined by θ. Call these loci $q(\theta)$ and $q(\theta)+1$, and note that the recombination fraction between them is given by $\phi_{q(\theta),q(\theta)+1}$. Index any putative trait location between these loci by $\lambda(\theta)$, the recombination fraction between the trait locus and marker locus $q(\theta)$. Then, again by assuming no interference, the recombination fraction between the trait locus and the marker locus $q(\theta)+1$ is given by

$$\lambda'(\theta) = \frac{\phi_{q(\theta),q(\theta)+1} - \lambda(\theta)}{1 - 2\lambda(\theta)} \tag{23}$$

Hence, leaving out some notational references to θ for clarity

$$P(I_l \mid H^t, \theta) = P\left(I_l \mid H^t_{l,q(\theta)}, H^t_{l,q(\theta)+1}, \lambda(\theta)\right) \tag{24}$$

$$= \lambda^{1-\delta(I_l, H^t_{l,q})}(1-\lambda)^{\delta(I_l, H^t_{l,q})}$$

$$\times \lambda'^{1-\delta(I_l, H^t_{l,q+1})}(1-\lambda')^{\delta(I_l, H^t_{l,q+1})} \tag{25}$$

where $\delta(i, j) = 1$ if $i = j$, and 0 otherwise.

Since this is simply a single locus collect evidence step, or single locus peeling, it again is a quick operation.

An added incentive to separate simulating inheritance states at the marker loci from evaluating the linkage of a particular trait is that the same simulations can be used for multiple trait models.

3. Results

As an illustration a family of 84 individuals spanning 5 generations was analysed. Fifty individuals for whom blood samples were available were genotyped at 26 markers spanning chromosome 14. We removed a marker near the centre of the map from the marker set and used it to simulate a disease trait. The marker has 4 alleles of which the most common has a frequency of about 50%. Individuals who carry a copy of this common allele, either homozygously or heterozygously, were randomly designated as *affected* with probability 90%. Non-carriers were designated affected with probability 5% to simulate sporadic incidence of the disease. We then attempted to map this pseudo-disease using only the affectedness status of the typed individuals and the observed genotypes at the remaining 25 markers.

To validate our Markov chain Monte Carlo method we ran an implementation of it, which we call *MCLINK*, on subsets of the data for which exact computations by GENEHUNTER and VITESSE were possible. A single iteration of MCLINK is defined by randomly updating first single locus blocks, the number of such updates being equal to the number of loci as described in 2.4, and then multilocus blocks for sets of individuals, as described in 2.5. For the GENEHUNTER comparison we used all 25 markers, but broke the family up into 11 informative nuclear families consisting of a total of 59 people. For the VITESSE comparison we used the whole pedigree but only 4 markers, the closest 2 on each side of the true location of the dropped out marker. We then ran MCLINK on the full pedigree with all 25 markers. The resulting Lod functions are displayed in Fig. 7. In each case the model assumed in calculation corresponded to

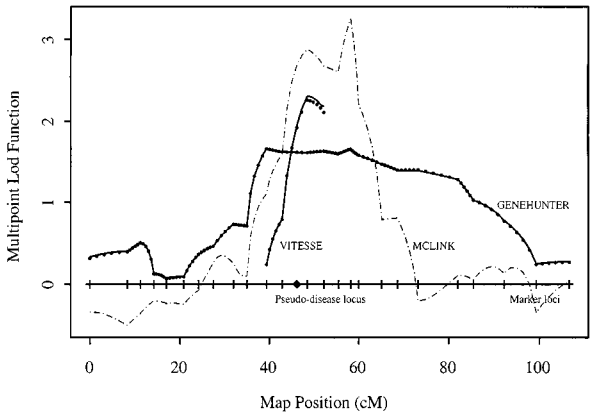

Fig. 7. *A comparison of MCLINK with GENEHUNTER and VITESSE. The solid line marked GENEHUNTER gives the resulting Lod function from running the program on nuclear families in the pedigree as if they were independent. The solid line marked VITESSE gives the resulting Lod function from running the entire pedigree but only for the 4 markers nearest the true gene location. Near each of the solid lines are dots representing the result of running MCLINK on the corresponding subsets of the data. The dashed line gives the Lod function calculated as an average of 5 MCLINK runs.*

the model used to simulate the data. This is an unrealistic best case, but is still valid for method comparison. For each calculation the simulation method was run for 2000 iterations for each of 5 independent seeds. As the starting configuration is chosen randomly, using a different seed determines a different starting point. Of the 2000 iterations, the first 1000 were discarded and the next 1000 were used for computation. All runs led to similar distributions of Lods across the region with very little variation. Although running simulations from different starting points can identify situations where there may be problems of near reducibility it cannot guarantee that none exist and other diagnostics would be helpful.

The simulation method approximates both the GENEHUNTER and VITESSE methods closely when the relevant partial data sets are used. For the full dataset, the Lod score reaches a peak very near the true location of the pseudo-disease locus. A *1-unit-down* support interval, roughly equivalent to a 90% confidence interval (Ott 1989), extends from 44cM to 60cM a region that contains the true location. By comparison the corresponding support interval derived from the 11 informative nuclear families considered disconnected extends from around 35cM to 90cM reflecting a substantial loss of precision. The fully informative Lod score at the true location of the pseudo disease is about 0.5 higher than that obtained by considering only 4 markers and about 1.2 higher than that obtained by splitting the family. Since all the statistics have the same distribution under the hypothesis of no linkage, 2 log(10) times the Lod score is approximately a χ^2_1, the type 1 errors corresponding to these maxima are 0.0003, 0.001 and 0.005. Clearly, there is a considerable disadvantage to either splitting up a single large pedigree or using only a subset of the markers available particularly with the multiple testing problem inherent in a whole genome search.

The computations were made on a SUN Ultra 60 workstation with a 300 MHz CPU. The VITESSE run required 24 seconds, the matching MCLINK run on 84 individuals by 4 markers took 1129 seconds. The GENEHUNTER run required 19 seconds, the matching MCLINK run on 59 individuals by 25 markers took 4938 seconds. The MCLINK run on the full dataset, 84 individuals by 25 markers, took 8903 seconds. The respective times for 10000 iterations for the three MCLINK runs are 3.36, 3.28 and 4.24 seconds per person per marker. The difference between the third and first is explained by varying numbers of alleles at the different markers. The difference between the third and second is explained by different genotyping density; the 25 individuals excluded from the second run, but included in the third are mostly ungenotyped ancestors who link up the three generation subfamilies. Very few allelic exclusions are possible for these individuals so they are not subject to the marginal gains in speed described in 2.6.

4. Discussion

4.1. *Mixing properties*

The guarantee of theoretical irreducibility that simultaneously changing all the variables for a locus in a single Gibbs update provides makes the method we develop here fundamentally different to current Markov chain Monte Carlo methods for linkage analysis. In practice, however, this theoretical property is no more important than updating subsets of inheritance states for some individuals across all the loci simultaneously. We believe that ours is the first method to use such multilocus updates. We also believe that this combination of sweeping updates of rows and columns of the inheritance matrix makes the mixing properties of our method superior to any existing method.

Our experience in running this program has been that the variance in Lod functions between different simulations for the same problem is unaffected by the number of loci considered. However, we still find examples of pedigrees in which we can, by running simulations from several starting points, detect near reducibility in the Markov chain. These pedigrees are usually large, over 150 people, and usually have many ungenotyped or partially genotyped individuals, making the state space that has to be sampled large and nearly disconnected. In many cases we have overcome the problems with new types of updates changing larger sets of rows of the inheritance matrix. The update sets described in Section 2.5 were developed from such problem cases and we expect such developments to continue. The density of marker loci is also a factor. Closely spaced markers define the true underlying states more accurately and give better convergence performance. Several of the large families which diverged with sparse marker sets have yielded consistent results when additional loci in the region were typed.

As is clearly shown in our example this is a case where approximate computation of the exact function gives a more powerful statistical method than exact computation of approximate functions because large pedigrees can be analysed intact.

4.2. *Programs*

The running times for the MCLINK program are considerably longer than the times needed for exact computation when it is possible. However, it should be noted that any one of the 5 individual simulation runs would, in each case, have given results differing negligibly from the mean. MCLINK is fast enough to use routinely in whole genome searches for linkage in large families. We have begun reviewing our programs and made several improvements in speed since the results in Section 3 were produced. Our next version will run approximately 5 times faster.

Although the method we describe here applies to pedigrees of arbitrary complexity, our implementation has been for the particular family structures that we usually analyse—mostly unlooped. As our programs rely on our underlying databases and consistency checking programs they are not currently suitable for distribution. The next step should be to implement the method in full generality. One good starting point would be the additions that Clauss Skaanning (Jensen 1997) made to the HUGIN (Andersen *et al.* 1989, Fischer 1990) programs. These implement block updating for general Bayesian networks and might be amended to use the block structures we develop here and to exploit the particular computational savings possible in genetic application. Another starting point might be the BUGS (Thomas *et al.* 1992) program.

4.3. *Other possible updating schemes*

In implementations on very complex pedigrees, updating all the variables at a locus simultaneously may not be possible, or desirable, if it takes too long. In such situations we could substitute updates on several overlapping subsets of the variables whose union is the set of all the variables at the locus. In much the same way as Lange and Elston (1975) dealt with loops by conditioning on one member of a loop, we can create several blocks which leave out a different member of the loop each time. While such schemes may no longer guarantee irreducible Markov chains, block updating seems to have very good mixing properties in practice (Jensen 1997).

Some more consideration should also be given to schemes that update the rows of the inheritance matrix. It may well be possible, by exploiting the pedigree structure, to choose particular subsets that can be updated more quickly than the general cases we outline above.

It might also be worth considering the original approach of Jensen (1997), which seeks to maximise the number of variables updated in each iteration without inspection of the particular structure. As these updates can't necessarily guarantee irreducibility, they should be used in addition to, and not instead of, our single locus updates.

4.4. *Modelling issues*

We have used simulations for parametric linkage analysis. They could also be used for non-parametric statistics that rely on identity by descent states determined by the inheritance vector. This involves simply substituting a sample of inheritance vectors for a complete enumeration as a method of approximation. The statistics described by Kruglyak *et al.* (1996), based on the scoring functions of Whittemore and Halpern (1994), are suitable candidates for this application.

With simulation, a broader range of more complex models can become feasible. For example, models that require two different genes in two different genetic regions to determine disease susceptibility can be addressed. Conditioning on a particular simulated inheritance matrix, this is a two locus problem which is tractable. While this is probably too intensive to apply to all possible pairs of genomic regions in an exhaustive genomic search, it could be a valuable tool for disentangling interaction between candidate genes.

4.5. *Conclusion*

We have described a Markov chain Monte Carlo method for sampling historical inheritance data for which the computational requirements for each iteration grow linearly both with the number of people and number of markers considered. This makes whole chromosome multilocus linkage analysis possible on large extended families for the first time. The method is widely applicable in classical and Bayesian frameworks, with highly parameterised modelling and with non-parametric approaches which depend on identity by descent states.

We have also tried to place the genetic problem more firmly in the broader context of graphical modelling and feel strongly that a more widespread understanding of this connection can yield positive results for both the special and the general case.

References

Andersen S.K., Olesen K.G., Jensen F.V., and Jensen F. 1989. HUGIN—A shell for building belief universes for expert systems. In: Proceedings of the 11th International Joint Conference on Artificial Intelligence, pp. 1080–1085.

Cannings C., Thompson E.A., and Skolnick M.H. 1978. Probability functions on complex pedigrees. Annals of Applied Probability 10: 26–61.

Cottingham R.W., Idury R.M., and Schaffer A.A. 1993. Faster sequential genetic linkage computations. American Journal of Human Genetics 53: 252–263.

Dawid A.P. 1992. Applications of a general propogation algorithm for probabilistic expert systems. Statistics and Computing 2: 25–36.

Dempster A.P., Laird N.M., and Rubin D.B. 1977. Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion). Journal of the Royal Statistical Society, Series B 39: 1–38.

Elston R.C. and Stewart J. 1971. A general model for the genetic analysis of pedigree data. Human Heredity 21: 523–542.

Fischer L.P. 1990. Reference manual for the HUGIN Application Programming Interface. Technical Report, Hugin Expert A/S.

Gelfand A.E. and Smith A.F.M. 1990. Sampling-based approaches to calculating marginal densities. Journal of the American Statistical Association 85(410): 398–409.

Gelman A. and Rubin D.B. 1992. Inference from iterative simulation using single and multiple sequences (with discussion) 7: 457–511.

Geman S. and Geman D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 45: 721–741.

Geyer C.J. 1992. Practical Markov chain Monte Carlo. Statistical Science 7: 156–163.

Harbron C. and Thomas A. 1994. Alternative graphical representations of genotypes in a pedigree. IMA Journal of Mathematics Applied in Medicine and Biology 11: 217–228.

Heath S.C. 1998. Generating consistent genotypic configurations for multi-allelic loci and large complex pedigrees. Human Heredity 48: 1–11.

Jensen C.S. 1997. Blocking Gibbs sampling for inference in large and complex Bayesian networks with applications in genetics. PhD Thesis, Department of Computer Science, Institute for Electronic Systems, Aalborg University, Denmark.

Jensen C.S. 1998. A simple method for finding a legal configuration in complex bayesian networks. Statistics and Computing 8(3): 243–251.

Jensen C.S. and Kong A. 1996. Blocking Gibbs sampling for linkage analysis in large pedigrees with many loops. Technical Report R-96-2048, Department of Computer Science, Aalborg University.

Jensen C.S., Kong A., and Kjaerulff U. 1995. Blocking-Gibbs sampling in very large probabilistic expert systems. International Journal of Human-Computer Studies 42: 647–666.

Jensen F.V. 1988. Local computation with probabilities on graphical structures and their application to expert systems, discussion contribution. Journal of the Royal Statistical Society, Series B 50: 157–224.

Jensen F.V. 1996. An Introduction to Bayesian Networks. UCL Press.

Kong A. 1991. Efficient methods for computing likage likelihoods of recessive diseases in inbred pedigrees. Genetic Epidemiology 8: 81–103.

Kong A., Cox N., Frigge M., and Irwin M. 1993. Sequential imputation and multipoint linkage analysis. Genetic Epidemiology 10: 483–488.

Kruglyak L., Daly M.J., Reeve-Daly M.P., and Lander E.S. 1996. Parametric and non-parametric linage analysis: A unified multipoint approach. American Journal of Human Genetics 58: 1347–1363.

Kruglyak L. and Lander E.S. 1998. Faster mulitipoint lankage analysis using Fourier transforms. Journal of Computational Biology 5(1): 1–7.

Lander E.S. and Green P. 1987. Construction of multilocus genetic maps in humans. Proc. Natl. Aca. Sci. USA 84: 2363–2367.

Lange K. and Elston R.C. 1975. Extensions to pedigree analysis. I. Likelihood calculations for simple and complex pedigrees. Human Heredity 25: 95–105.

Lathrop G.M., Lalouel J.M., Julier C., and Ott J. 1984. Strategies for multilocus linkage analysis in humans. Proc. Natl. Acad. Sci. USA 81: 3443–3446.

Lauritzen S.L. 1996. Graphical Models. Clarendon Press.

Lauritzen S.L. and Spiegelhalter D.J. 1988. Local computations with probabilities on graphical structures and their applications to expert systems. Journal of the Royal Statistical Society, Series B 50: 157–224.

Morton N.E. 1955. Sequential tests for the detection of linkage. American Journal of Human Genetics 7: 23–30.

O'Connell J.R. and Weeks D.E. 1995. The VITESSE algorithm for rapid exact multilocus linkage analysis via genotype set-recoding and fuzzy inheritance. Nature Genetics 11: 402–408.

Ott J. 1989. Computer simulation methods in human linkage analysis. Proc. Natl. Acad. Sci. USA 86, 4175–4178.

Ploughman L.M. and Boehnke M. 1989. Estimating the power of a proposed linkage study for a complex genetic trait. American Journal of Human Genetics 44: 543–551.

Sheehan N. 1990. Genetic restoration on complex pedigrees. Ph.D. Thesis, University of Washington.

Sheehan N. 1992. Sampling genotypes on complex pedigrees with phenotypic constraints: The origin of the B allele among the Polar Eskimos. IMA Journal of Mathematics Applied in Medicine and Biology 9: 1–18.

Sheehan N. and Thomas A. 1993. On the irreducibility of a Markov chain defined on a space of genotype configurations by a sampling scheme. Biometrics 49: 163–175.

Smith A.F.M. and Roberts G.O. 1993. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. Journal of the Royal Statistical Society, Series B 55(1): 5–23.

Sobel E. and Lange K. 1996. Descent graphs in pedigree analysis: Applications to haplotyping, location scores, and marker-sharing statistics. American Journal of Human Genetics 58: 1323–1337.

Thomas A. 1986. Optimal computation of probability functions for pedigree analysis. IMA Journal of Mathematics Applied in Medicine and Biology 3: 167–178.

Thomas A. 1994. Linkage analysis on complex pedigrees by simulation. IMA Journal of Mathematics Applied in Medicine and Biology 11: 79–93.

Thomas A., Spiegelhalter D.J., and Gilks W.R. 1992. BUGS: A program to perform Bayesian inference using Gibbs sampling. In: Bernardo J.M., Berger J.O., Dawid A.P., and Smith A.F.M. (Eds.), Bayesian Statistics, Vol. 4. Oxford, Clarendon Press, pp. 837–842.

Whittemore A.S. and Halpern J. 1994. A class of tests for linkage using affected pedigree members. Biometrics 50: 118–127.