Genet. Sel. Evol. 33 (2001) 337–367

Original article

Sampling genotypes in large pedigrees with loops

Soledad A. Fernández[a,b], Rohan L. Fernando[a,c,*], Bernt Guldbrandtsen[d], Liviu R. Totir[a], Alicia L. Carriquiry[b,c]

[a] Department of Animal Science, Iowa State University,
225 Kildee Hall, Ames, IA 50011, USA
[b] Department of Statistics, Iowa State University,
225 Kildee Hall, Ames, IA 50011, USA
[c] Lawrence H. Baker Center for Bioinformatics and Biological Statistics,
Iowa State University, Ames, IA 50011, USA
[d] Danish Institute of Animal Science, Foulum, Denmark

(Received 19 October 2000; accepted 23 February 2001)

Abstract – Markov chain Monte Carlo (MCMC) methods have been proposed to overcome computational problems in linkage and segregation analyses. This approach involves sampling genotypes at the marker and trait loci. Scalar-Gibbs is easy to implement, and it is widely used in genetics. However, the Markov chain that corresponds to scalar-Gibbs may not be irreducible when the marker locus has more than two alleles, and even when the chain is irreducible, mixing has been observed to be slow. These problems do not arise if the genotypes are sampled jointly from the entire pedigree. This paper proposes a method to jointly sample genotypes. The method combines the Elston-Stewart algorithm and iterative peeling, and is called the ESIP sampler. For a hypothetical pedigree, genotype probabilities are estimated from samples obtained using ESIP and also scalar-Gibbs. Approximate probabilities were also obtained by iterative peeling. Comparisons of these with exact genotypic probabilities obtained by the Elston-Stewart algorithm showed that ESIP and iterative peeling yielded genotypic probabilities that were very close to the exact values. Nevertheless, estimated probabilities from scalar-Gibbs with a chain of length 235 000, including a burn-in of 200 000 steps, were less accurate than probabilities estimated using ESIP with a chain of length 10 000, with a burn-in of 5 000 steps. The effective chain size (ECS) was estimated from the last 25 000 elements of the chain of length 125 000. For one of the ESIP samplers, the ECS ranged from 21 579 to 22 741, while for the scalar-Gibbs sampler, the ECS ranged from 64 to 671. Genotype probabilities were also estimated for a large real pedigree consisting of 3 223 individuals. For this pedigree, it is not feasible to obtain exact genotype probabilities by the Elston-Stewart algorithm. ESIP and iterative peeling yielded very similar results. However, results from scalar-Gibbs were less accurate.

genotype sampler / Markov chain Monte Carlo / peeling

* Correspondence and reprints
E-mail: rohan@iastate.edu

1. INTRODUCTION

Probability functions such as likelihood functions and genotype probabilities play an important role in the analysis of genetic data. For example, likelihoods given genotypic and phenotypic data are needed in segregation and linkage analyses. In genetic evaluations, conditional genotype probabilities are used to compute conditional means of genotypic values. These conditional means are then used to rank individuals for selection. Conditional genotype probabilities are also used in genetic counseling. For example, in the case of recessive disease traits it is important to know which individuals in a population are probable carriers of a deleterious allele.

When inheritance is monogenic and the pedigree has no loops, the likelihood can be computed efficiently using the Elston-Stewart algorithm [3], which is also called "peeling." For small pedigrees (about 100 members) with loops, extensions of the Elston-Stewart algorithm have been developed for evaluating the likelihood [2,24,25,28,29]. These methods were developed in human genetics. In livestock, pedigrees are usually much larger and contain many more loops. Thus, the application of computer-intensive methods developed for humans will often be difficult or inappropriate in livestock data.

Van Arendonk *et al.* [36] presented an iterative algorithm to calculate genotype probabilities for all members in an animal pedigree. Some limitations in their algorithm were removed by Janss *et al.* [21]. Their method can be used to approximate the likelihood for large and complex pedigrees with loops. Stricker *et al.* [27] also proposed a method to approximate the likelihood in pedigrees with loops. This method is based on an algorithm that cuts the loops. In 1996, Wang *et al.* proposed a new approximation to the likelihood of a pedigree with loops by cutting all loops and extending the pedigree at the cuts. This method makes use of iterative peeling. They showed that the likelihood computed by iterative peeling is equivalent to the likelihood computed from a cut and extended pedigree.

It is not straightforward to calculate the exact pedigree likelihood under mixed inheritance [1,7,13,14]. The reason is that phenotypic values of pedigree members cannot be assumed to be conditionally independent, given only the major genotypes of the pedigree members, because the phenotypic value is also influenced by the polygenic loci. Alternative models have been adopted to overcome this problem [1,7]. Bonney [1] proposed a regressive model where conditional covariances between relatives, given the major genotypes, are modeled directly through the phenotypes. Thus, this model is not suitable for pedigrees with a large proportion of missing phenotypic values. Fernando *et al.* [7] presented a finite polygenic mixed model that has the advantage that its likelihood can be calculated using efficient algorithms developed for oligogenic models. A disadvantage of this approach is that it cannot accommodate nongenetic covariances among relatives. Hasstedt [13,14] has used approximations for computing the likelihood. The approximation proposed in 1991 accommodates a completely general structure for the nongenetic residual covariances. But under this approach, the phenotypic covariance matrix must be inverted to compute the likelihood. This makes the approximation very inefficient for large pedigrees. Furthermore, the accuracy of the method cannot be determined when it is implemented in large pedigrees.

Markov chain Monte Carlo (MCMC) methods have been proposed to overcome these problems. These MCMC methods can be used to obtain estimates to any desired level of accuracy. As Thomas and Cortessis [30] observed, the genotypes in a pedigree are sampled according to a Markovian process, because a neighborhood system can be defined on a pedigree such that the genotype of an individual, conditional on the neighbors (or relatives), is independent of the remaining pedigree members. This local dependency makes MCMC methods, such as the Gibbs sampler, very easy to implement and provides a strategy to sample genotypes from the joint posterior distribution of genotypes [26]. The samples are used either in maximum likelihood [12,31,32] or Bayesian methods [17–20,30,35] for segregation or linkage analysis.

When using the Gibbs sampler, however, mixing can be very slow due to the "vertical dependence" between genotypes of parents and progeny [20]. The larger the progeny groups, the stronger the dependence, and thus the Gibbs chains do not move. Poor mixing has also been encountered due to the "horizontal dependence" between genotypes at tightly linked loci [34]. When this happens it is said that the chains are reducible "in practice."

The problem of poor mixing due to vertical dependence can be reduced by jointly sampling blocks of genotypes at a single locus [20,23]. In this approach, the blocks are typically formed by subfamilies in the pedigree. The efficiency of blocking depends on the pedigree structure and the way those blocks are built. Further, the scalar-Gibbs chains may not be irreducible when sampling genotypes at marker loci with more than two alleles [26,30]. By blocking Gibbs, this problem is expected to be reduced, but is not guaranteed to be eliminated [22]. The problem of poor mixing due to horizontal dependence can be reduced by sampling blocks of the tightly linked genotypes jointly within an individual [33,34]. However, with extended pedigrees poor mixing may still be a problem, and further, this sampler is not guaranteed to be irreducible when sampling genotypes at multi-allelic loci.

It has been proposed to extend the idea of blocking Gibbs to sample genotypes jointly at a single locus from the *entire* pedigree in such a way that irreducibility is guaranteed [5]. The proposed sampler is based on the Elston-Stewart algorithm and iterative peeling, and so it will be referred to as the ESIP sampler. To study the mixing performance of the ESIP sampler at a single locus, it was first applied to the relatively simple problem of sampling genotypes at a biallelic disease locus. This paper documents the results from this study. The mixing performance of ESIP for sampling genotypes at tightly linked loci has not been examined yet. Given the positive results that were obtained in this study, the performance of the sampler is currently being evaluated for sampling missing genotypes at a marker locus with more than two alleles. A manuscript with a detailed proof of the irreducibility of the sampler and results from the second study is under preparation.

In brief, genotypes are jointly sampled as follows. When there are no loops or when the pedigree contains only "simple" loops, we first peel the entire pedigree using the Elston-Stewart algorithm (exact peeling). Then, genotypes are sampled by "reverse peeling" [16,20,23]. When the loops are complex and exact peeling cannot be undertaken efficiently, we obtain a joint sample from a pedigree that is modified to make peeling efficient. This sample is used in the Metropolis-Hastings algorithm to obtain draws from the unmodified pedigree. The modification that we use involves cutting some of the loops as in Stricker *et al.* and extending the pedigree at the cuts as in Wang *et al.* [37]. The "cutting" and "extension" of the pedigree is not done explicitly but is done instead by "iterative peeling."

On the one hand, although exact peeling of pedigrees with loops is not new in human genetics, it is relatively new in livestock applications. On the other hand, iterative peeling was introduced in livestock applications to obtain approximate probabilities for complex pedigrees. In this paper, these two approaches are combined for sampling genotypes in complex pedigrees. Therefore, for completeness, in Section 2, we explain how genotypes can be sampled efficiently by exact peeling for a pedigree with simple loops. In Section 3, we explain how genotypes can be sampled efficiently by iterative peeling for a pedigree with complex loops. In Section 4, we describe how exact and iterative peeling can be combined to improve the efficiency of the sampler.

Finally, in Section 5, the ESIP sampler is evaluated by computing genotype probabilities for a monogenic trait in a small hypothetical pedigree and in a large real pedigree. This section also includes the evaluation of iterative peeling.

2. EXACT PEELING TO SAMPLE GENOTYPES

Consider the pedigree shown in Figure 1. We introduce some notation, and show how exact peeling can be used to sample genotypes in this pedigree. Let $\mathbf{g}$ be the vector of genotypes and $\mathbf{y}$ be the vector of phenotypes in this pedigree.

To obtain a random sample from $f(\mathbf{g}|\mathbf{y})$, we can use a rejection sampler [9] based on $f(\mathbf{g}|\mathbf{y})$, but this may be very inefficient.

Instead, we sample individuals sequentially as described below. To obtain a sample from $f(g_1, g_2, g_3, g_4, g_5, g_6, g_7|\mathbf{y})$ in Figure 1, we first sample the

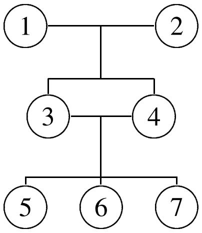


Figure 1. Simple two-generational pedigree with loop.

genotype for individual 1 from $f(g_1|\mathbf{y})$. Next we sample g_2 from $f(g_2|g_1, \mathbf{y})$, g_3 from $f(g_3|g_1, g_2, \mathbf{y})$, and so on. To compute $f(g_1|\mathbf{y})$ we use peeling [2,3]. The first step in computing $f(g_1|\mathbf{y})$ is to compute the likelihood of the pedigree.

The likelihood for the pedigree in Figure 1 can be written as

$$L \propto \sum_{g_1} \sum_{g_2} \cdots \sum_{g_7} h(g_1)h(g_2)h(g_1, g_2, g_3)h(g_1, g_2, g_4)h(g_3, g_4, g_5) \times h(g_3, g_4, g_6)h(g_3, g_4, g_7) \quad (1)$$

where $h(g_j) = P(g_j)f(y_j|g_j)$, $f(y_j|g_j)$ is the probability that an individual with genotype g_j has phenotype y_j (penetrance function), $P(g_j)$ is the marginal probability that an individual has genotype g_j (founder probability), $h(g_m, g_f, g_j) = P(g_j|g_m, g_f)f(y_j|g_j)$, g_m and g_f are the genotypes for the mother and father of individual j, and $P(g_j|g_m, g_f)$ is the probability that an individual has genotype g_j given parental genotypes g_m and g_f (transition probability).

Suppose each g_j can take on one of three values (AA, Aa, and aa). Then L as given in (1) is the sum of 3^7 terms, and the number of computations is exponential in the number of individuals in the expression. Thus, directly computing the likelihood as given in (1) is feasible only for small pedigrees. The Elston-Stewart algorithm [3], however, provides an efficient method to compute (1) for pedigrees without loops, and generalizations of this algorithm [2,24,25] provide strategies to compute the likelihood efficiently for general pedigrees with simple loops.

Consider the summation over g_7. In (1) this summation is done for all combinations of values of g_1, g_2, g_3, g_4, g_5, and g_6. However, the only function involving g_7, is $h(g_3, g_4, g_7)$, which depends only on two other individual genotypes (g_3 and g_4). In the Elston-Stewart algorithm the summation over g_7 is done only for all combinations of values of g_3 and g_4. The results from this summation are stored in a two-dimensional table, $c_7(g_3, g_4)$, called a *cutset*:

$$c_7(g_3, g_4) = \sum_{g_7} h(g_3, g_4, g_7).$$

After summing out g_7 and reordering equation (1), the likelihood is written as

$$L \propto \sum_{g_1} \sum_{g_2} h(g_1)h(g_2) \sum_{g_3} h(g_1, g_2, g_3) \sum_{g_4} h(g_1, g_2, g_4)c_7(g_3, g_4) \\ \times \sum_{g_5} h(g_3, g_4, g_5) \sum_{g_6} h(g_3, g_4, g_6). \quad (2)$$

Now, we can sum out g_6. The only function involving g_6 in (2) is $h(g_3, g_4, g_6)$, which also depends on the genotypes of individuals 3 and 4. Thus, the summation is done for all combinations of values of g_3 and g_4 and the results are stored in $c_6(g_3, g_4)$:

$$c_6(g_3, g_4) = \sum_{g_6} h(g_3, g_4, g_6).$$

This process is continued until all individuals have been summed out. Computing L sequentially as described above is referred to as *peeling*. In the first step, g_7 was *peeled*, and a simpler expression was obtained that did not involve g_7. Similarly, after peeling g_6, L becomes free of g_6. To compute L efficiently, the order of peeling is critical. For example, consider peeling g_1 as the first step, so the likelihood can be written as

$$L \propto \sum_{g_2} \sum_{g_3} \cdots \sum_{g_7} h(g_2)h(g_3, g_4, g_5)h(g_3, g_4, g_6)h(g_3, g_4, g_7)c_1(g_2, g_3, g_4)$$

where

$$c_1(g_2, g_3, g_4) = \sum_{g_1} h(g_1)h(g_1, g_2, g_3)h(g_1, g_2, g_4).$$

The result, $c_1(g_2, g_3, g_4)$, from peeling g_1 is a cutset of size 3, and its computation involves summing over g_1 for all genotype combinations of g_2, g_3, and g_4. Computing $c_7(g_3, g_4)$ has lower storage and computational requirements than computing $c_1(g_2, g_3, g_4)$. The storage and computational requirements would be similar for peeling g_5 and g_6 in the first step. Peeling g_3 or g_4 in the first step would be even more costly, in terms of computational requirements, than peeling g_1 or g_2 first.

Thus, to evaluate the likelihood for this pedigree we first need to determine the peeling order. Following [24], the peeling order is determined by the algorithm described below.

1. List all the individuals in the pedigree that need to be peeled.
2. For each individual determine the size of the resulting cutset after peeling that individual.

3. Peel the individual with the smallest cutset.
4. Repeat steps 2 and 3 until all individuals are peeled.

In this case, an efficient peeling order is: 7, 6, 5, 4, 3, 2, and 1.

Determining an optimal peeling order is related to the problem of solving systems of symmetric sparse linear equations [4]. When Gaussian elimination is used to solve such equations, some coefficients that were initially zero become nonzero, *i.e.*, get "filled in". The number of coefficients that get filled in depends on the order of elimination. Much research has been conducted in this area, and sophisticated algorithms have been developed to determine the order to minimize the number of coefficients that get filled in at each step. It can be shown that determining an optimal peeling order is equivalent to determining an optimal order of elimination in sparse system of linear equations. Thus algorithms that have been developed to determine the order of elimination in sparse linear systems can also be used to determine peeling order [4]. Once we establish a peeling order, we can represent the operations involved in the peeling process as shown in Table I. The first column in this table gives the peeling sequence. The subsequent columns give the factors in the likelihood at different stages of peeling.

Before peeling any individuals, the seven factors in the likelihood (1) are represented in the second column of Table I. For example, (3, 4, 7) in the first row represents the factor $h(g_3, g_4, g_7)$ in equation (1), and (2) in the 6th row of Table I represents $h(g_2)$ in equation (1). In this table, cutsets are represented as {.,.}.

After peeling 7, a cutset involving genotypes of individuals 3 and 4 is generated, $c_7(g_3, g_4)$, and it is represented as {3,4} in the third column of Table I. Any cutset that results from peeling an individual becomes a factor in the row of the first individual in the cutset to be peeled. Thus, in this example, cutset {3,4} becomes a factor in the row of individual 4, since 4 is peeled before 3.

Next, when we peel 6, the cutset $c_6(g_3, g_4)$ is generated, and it becomes a new factor in the row of individual 4. Thus, it is represented in the fourth column of Table I as a second set {3,4}.

When we peel 5, $c_5(g_3, g_4)$ is generated, and it is represented as {3,4} in the row of individual 4 (fifth column of Table I). Next, we peel 4, and $c_4(g_1, g_2, g_3)$ is generated. This cutset becomes a factor in the row of individual 3 (sixth column of Table I). Next, we peel 3, and $c_3(g_1, g_2)$ becomes a factor in the row of individual 2 (seventh column of Table I). When we peel 2, $c_2(g_1)$ is generated, and it is represented as {1} in the row corresponding to individual 1 in the last column of Table I.

Peeling 1 results in the likelihood (L). Now, we sample genotypes in the reverse order in which they were peeled (reverse peeling; Heath [16]). In this

Table I. Peeling sequence and factors in the likelihood at different stages of peeling. Cutsets are indicated by { }.

Peeling	Factors in the likelihood after peeling individual j:						
sequence	$j = -$	$j = 7$	$j = 6$	$j = 5$	$j = 4$	$j = 3$	$j = 2$
7	(3,4,7)						
6	(3,4,6)	(3,4,6)					
5	(3,4,5)	(3,4,5)	(3,4,5)				
4	(1,2,4)	(1,2,4){3,4}	(1,2,4){3,4}{3,4}	(1,2,4){3,4}{3,4}{3,4}			
3	(1,2,3)	(1,2,3)	(1,2,3)	(1,2,3)	(1,2,3){1,2,3}		
2	(2)	(2)	(2)	(2)	(2)	(2){1,2}	
1	(1)	(1)	(1)	(1)	(1)	(1)	(1){1}

example, after peeling individual 1 we compute the marginal probability for 1 as

$$f(g_1|\mathbf{y}) = \frac{h(g_1)c_2(g_1)}{L}\cdot$$

Note that to compute $f(g_1|\mathbf{y})$ we are using the factors represented in the last row and column of Table I, *i.e.*, the numerator of this equation is the product of the factors in the 7th row of Table I. Once $f(g_1|\mathbf{y})$ has been obtained, we sample g_1 using the inverse cumulative probability function. Next, we compute

$$f(g_2|g_1, \mathbf{y}) = \frac{h(g_2)c_3(g_1, g_2)}{\sum_{g_2} h(g_2)c_3(g_1, g_2)}$$

and then we sample g_2 from $f(g_2|g_1, \mathbf{y})$. Again, the factors involved in the computation of $f(g_2|g_1, \mathbf{y})$ are represented in the 6th row of Table I. Thus, the factors needed to sample g_i are those used in peeling i.

By applying this sampling procedure, we eventually generate a sample from the joint distribution of all genotypes for the entire pedigree. The sampling sequence in this case is:
sample g_1 from $f(g_1|\mathbf{y})$,
sample g_2 from $f(g_2|\mathbf{y}, g_1)$,
sample g_3 from $f(g_3|\mathbf{y}, g_1, g_2)$,
sample g_4 from $f(g_4|\mathbf{y}, g_1, g_2, g_3)$,
sample g_5 from $f(g_5|\mathbf{y}, g_1, g_2, g_3, g_4)$,
sample g_6 from $f(g_6|\mathbf{y}, g_1, g_2, g_3, g_4, g_5)$,
sample g_7 from $f(g_7|\mathbf{y}, g_1, g_2, g_3, g_4, g_5, g_6)$.

In pedigrees with complex loops, peeling methods as described above are not feasible. The reason is that the cutsets generated after peeling some individuals become too large when there are complex loops in the pedigree.

3. ITERATIVE PEELING TO SAMPLE GENOTYPES

Exact peeling methods cannot be applied when pedigrees are large and have complex loops. Iterative peeling [6,21,36,37], however can be used to get approximate results. To describe iterative peeling we use a small pedigree with a simple loop, which is presented as a directed graph (Fig. 2(a)).

Before peeling, the graph contains individual nodes and mating nodes. Each individual node is indicated by the individual identification number; they correspond to the penetrance functions, and in the case of founders, also include the founder probability function. Each mating node is indicated by an oval, which corresponds to the transition probability function. The edges in the graph connect the mating nodes with the parents and with the offspring.

Before proceeding with iterative peeling we modify the graph by merging mating nodes into nuclear-family nodes. The resulting graph with the merged

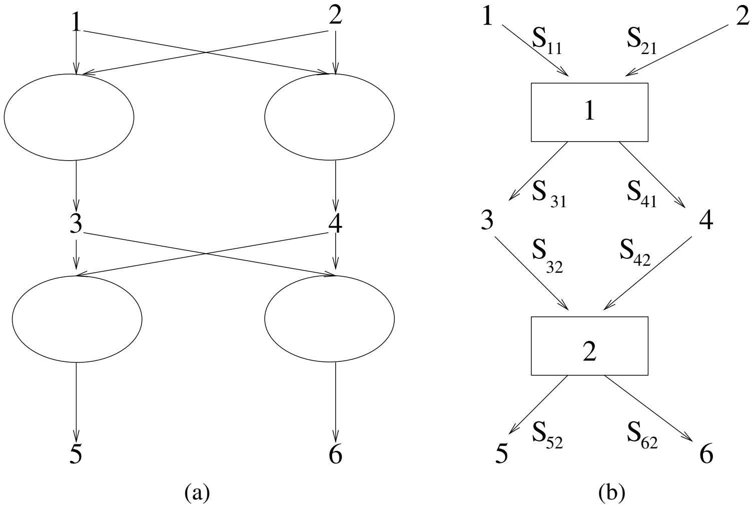


Figure 2. Graph representation of a two-generational pedigree with loops.

mating nodes is shown in Figure 2(b). Here, the nuclear-family nodes are represented by rectangles. There are eight edges: S_{11}, S_{21}, S_{31}, S_{32}, S_{41}, S_{42}, S_{52}, and S_{62} in this graph. The first subindex of S indicates the individual number, and the second subindex indicates the nuclear-family node number; for example S_{31} is the edge that connects individual 3 with nuclear-family node 1.

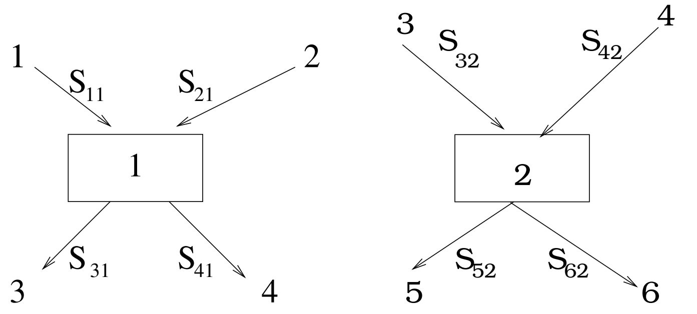
The edge between a parent and a nuclear-family has been called a "posterior" probability, and the edge between an offspring and a nuclear-family has been called an "anterior" probability [6,37]. In the next section, iterative peeling will be combined with exact peeling of pedigrees with loops. Then, there will be edges between individuals and cutsets. In this section, iterative peeling is reformulated such that, in the next section, it can be extended to accommodate edges between individuals and cutsets. We use this small example to explain iterative peeling and present general expressions for the algorithm later.

Suppose we want to sample the genotype for individual 1 from $f(g_1|\mathbf{y})$. We first obtain an estimate for the edge probability S_{11}, connecting individual 1 to the rest of the pedigree through nuclear family 1. Once S_{11} is computed, the genotype probabilities are computed as

$$f(g_1|\mathbf{y}) = \frac{f(y_1|g_1)P(g_1)S_{11}}{\sum_{g_1} f(y_1|g_1)P(g_1)S_{11}}. \tag{3}$$

Below we describe how to iteratively compute S_{11}.

We first initialize all the edge probabilities. In general, all edge probabilities are initialized to 1. For this example, however, it is convenient to set S_{41}



(a) (b)

Figure 3. Nuclear families.

equal to the founder genotype probabilities. Once the edges are initialized, we iteratively update edge probabilities using the phenotypes and the current values of the appropriate edges (explained below) of all the individuals in the corresponding nuclear family. Thus, we update S_{11} as

$$S_{11} = \sum_{g_2}\sum_{g_3}\sum_{g_4} f(y_2|g_2)P(g_2)f(y_3|g_3)P(g_3|g_1, g_2)f(y_4|g_4)P(g_4|g_1, g_2)S_{32}S_{42}.$$

At this stage, S_{11} is the conditional probability $f(y_2, y_3, y_4|g_1)$. The value of $f(y_2, y_3, y_4|g_1)$ is the conditional probability of the phenotypic values of individuals 2, 3 and 4 given the genotypic value of 1 in the pedigree shown in Figure 3(a). Note that the edges that contributed to updating S_{11} are those that connect the members of nuclear family 1 to other nuclear families.

Similarly, S_{21} is updated as

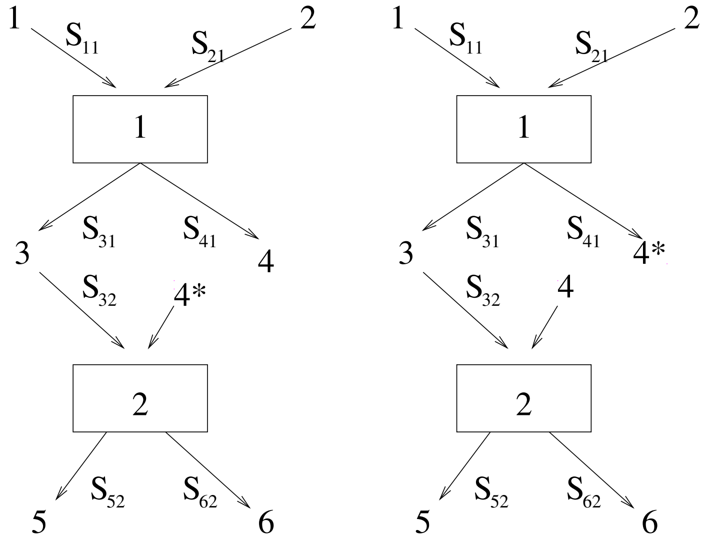
$$S_{21} = \sum_{g_1}\sum_{g_3}\sum_{g_4} f(y_1|g_1)P(g_1)f(y_3|g_3)P(g_3|g_1, g_2)f(y_4|g_4)P(g_4|g_1, g_2)S_{32}S_{42}.$$

and is the conditional probability $f(y_1, y_3, y_4|g_2)$. Next, we update S_{31} as

$$S_{31} = \sum_{g_1}\sum_{g_2}\sum_{g_4} \times f(y_1|g_1)P(g_1)f(y_2|g_2)P(g_2)P(g_3|g_1, g_2)f(y_4|g_4)P(g_4|g_1, g_2)S_{42}$$

which is the joint probability $f(y_1, y_2, y_4, g_3)$. Next, we update S_{32} as,

$$S_{32} = \sum_{g_4}\sum_{g_5}\sum_{g_6} f(y_5|g_5)P(g_5|g_3, g_4)f(y_6|g_6)P(g_6|g_3, g_4)f(y_4|g_4)\underbrace{S_{41}}_{P(g_4)}$$



(a) (b)

Figure 4. Cut and extended graphs.

which is the conditional probability $f(y_4, y_5, y_6|g_3)$ in the pedigree shown in Figure 3(b). Note that in these three cases, initial values of edge probabilities were used.

Next, we update S_{41} as

$$S_{41} = \sum_{g_1}\sum_{g_2}\sum_{g_3} \times f(y_1|g_1)P(g_1)f(y_2|g_2)P(g_2)f(y_3|g_3)P(g_3|g_1, g_2)P(g_4|g_1, g_2) \underbrace{S_{32}}_{f(y_4,y_5,y_6|g_3)} .$$

In this case, the edge probability S_{32} was already updated once. Thus, the value of $S_{41} = f(y_1, y_2, y_3, y_{4*}, y_5, y_6, g_4)$ is the joint probability of the genotype of individual 4 and of all the phenotypic values connected to 4 through nuclear family 1 in the cut-extended pedigree shown in Figure 4(a).

Next, we update S_{42} as

$$S_{42} = \sum_{g_3}\sum_{g_5}\sum_{g_6} f(y_5|g_5)P(g_5|g_3, g_4)f(y_6|g_6)P(g_6|g_3, g_4)f(y_3|g_3) \underbrace{S_{31}}_{f(y_1,y_2,y_4,g_3)} .$$

Again, in this case we use an edge probability that was already updated, and thus $S_{42} = f(y_1, y_2, y_3, y_{4*}, y_5, y_6|g_4)$, which is the conditional probability of all the phenotypic values connected to 4 through nuclear family 2 in the cut-extended pedigree shown in Figure 4(b), given the genotype of individual 4.

Each subsequent iteration results in further extensions to a cut pedigree. After a sufficient number of iterations we sample genotypes as follows from the iteratively peeled pedigree. First we sample the genotype of individual 1 from $f(g_1|\mathbf{y})$, which is computed using S_{11} as described above. Next, to sample the genotype of 2, we update S_{21} to reflect the sampled value for the genotype of 1 as

$$S_{21} = \sum_{g_3} \sum_{g_4} f(y_1|g_1)P(g_1)f(y_3|g_3)P(g_3|g_1, g_2)f(y_4|g_4)P(g_4|g_1, g_2)S_{32}S_{42}$$

where g_1 is the sampled value for the genotype of 1. Using this updated value for $S_{21}, f(g_2|\mathbf{y}, g_1)$ is computed as

$$f(g_2|\mathbf{y}, g_1) = \frac{f(y_2|g_2)P(g_2)S_{21}}{\sum_{g_2} f(y_2|g_2)P(g_2)S_{21}}$$

and g_2 is sampled. This process is continued until all individuals are sampled.

We now provide the general expressions for updating edge probabilities in iterative peeling and an algorithm for sampling genotypes. The edges between individuals and a nuclear family are updated taking advantage of the conditional independence of the offspring given their parents. The summations are done for all the individuals in the nuclear family except individual j. If individual j is a parent, the factors included in the summation are: the penetrance functions of all offspring and the spouse of individual j, the transition probabilities of all offspring of individual j, the founder probability of the spouse of j (if the spouse of individual j is a founder), and all the edges connecting all offspring and the spouse of j with other nodes. If individual j is an offspring, the factors included in the summation are: the penetrance function of all the siblings and parents of individual j, the founder probabilities of the parents (if the parents are founders), the transition probabilities of all the offspring in the nuclear family (but the summation is not done for individual j), and the edges connecting all the siblings and parents of j with other nodes.

Let S_{js} be an edge between individual j and nuclear-family node s. If j is a parent in the nuclear family, S_{js} is computed iteratively as

$$S_{js} = \sum_{g_p} R_{sp} \prod_{k \in C_s} \left[\sum_{g_k} \Pr(g_k|g_j, g_p)R_{sk} \right] \tag{4}$$

where p is the other parent in the nuclear family, and C_s is the set of children in nuclear family s,

$$R_{sl} = f(y_l|g_l)P(g_l)\left(\prod_{\substack{e\in E_l\\ e\neq s}} S_{le}\right) \tag{5}$$

for $l = k$ or p, E_l is the set of edges for individual $l, f(y_l|g_l)$ is the penetrance function, $P(g_l)$ is the founder probability if l is a founder, and $P(g_l) = 1$ if l is not a founder. If j is a child in the nuclear family, S_{js} is computed iteratively as

$$S_{js} = \Pr(g_j|g_m, g_f)\sum_{g_m,g_f} R_{sm}R_{sf}\prod_{\substack{k\in C_s\\ k\neq j}}\left[\sum_{g_k}\Pr(g_k|g_m, g_f)R_{sk}\right] \tag{6}$$

where m and f are the parents in the nuclear-family node. These definitions of edge probabilities are equivalent to the definitions of anterior and posterior probabilities used in Fernando *et al.*, 1993 [6]. For a pedigree without loops, these formulas converge to the exact probabilities.

All edge probabilities are iteratively updated using (4) and (6). After a sufficient number of iterations, we sample genotypes for all individuals in the pedigree. We start from an arbitrary individual and sample its genotype using the marginal probability function

$$f(g_j|y_j) = \frac{f(y_j|g_j)P(g_j)\prod_s S_{js}}{\sum_{g_j} f(y_j|g_j)P(g_j)\prod_s S_{js}} \tag{7}$$

where the product of S_{js} is over all edges for j. Then we sample a neighbor conditional on the sampled genotypes as follows. A neighbor is defined as any individual who is also a member of any nuclear-family node to which the sampled individual belongs. First, all edges of the individual to be sampled are updated to reflect the already sampled genotypes. To update edges, we use (4) and (6), but the summations are only over the unsampled genotypes. Now to sample the genotype conditional on the already sampled genotypes we use (7) with the edges that were updated for the sampled genotypes. After all genotypes are sampled, the Metropolis-Hastings step is used to accept or reject those sampled genotypes.

The Metropolis-Hastings acceptance probability [11] is

$$\eta = min\left(1, \frac{\pi(g_{\mathrm{c}})q(g_{\mathrm{prev}}|g_{\mathrm{c}})}{\pi(g_{\mathrm{prev}})q(g_{\mathrm{c}}|g_{\mathrm{prev}})}\right) \tag{8}$$

where π is the target distribution, q is the proposal distribution, g_{prev} is the accepted draw from the previous round, and g_{c} is the sampled candidate from the present round. The candidate sample g_{c} is accepted with probability η. We consider the special case of *independence sampling*: instead of $q(g_{\text{c}}|g_{\text{prev}})$ and $q(g_{\text{prev}}|g_{\text{c}})$, we use $q(g_{\text{prev}})$ and $q(g_{\text{c}})$. Thus

$$\eta = min\left(1, \frac{\pi(g_{\text{c}})q(g_{\text{prev}})}{\pi(g_{\text{prev}})q(g_{\text{c}})}\right). \tag{9}$$

The independence sampler can be used here because the proposal distribution is very close to the target, and thus the sampler can move far away from the neighborhood of the previous sample without increasing the rejection rate. We use the following expression to obtain $\pi(.)$ on the true pedigree,

$$\pi(\mathbf{g}) \propto \prod_{j=1}^{n_1} h(g_j) \prod_{j=n_1+1}^{n} h(g_j|g_{f_j}, g_{m_j}) \tag{10}$$

where g_{f_j}, g_{m_j} are the genotypes of the parents of individual j, and n_1 is the number of founders. In this example, $\pi(\mathbf{g})$ is

$$\pi(\mathbf{g}) \propto h(g_1)h(g_2)h(g_1, g_2, g_3)h(g_1, g_2, g_4)h(g_3, g_4, g_5)h(g_3, g_4, g_6).$$

To compute $q(.)$ we multiply the probabilities that were used in the sampling process described above. For example, for this pedigree $q(\mathbf{g})$ is

$$q(\mathbf{g}) = f(g_1|\mathbf{y})f(g_2|\mathbf{y}, g_1)\cdots f(g_6|g_1, g_2, g_3, g_4, g_5, \mathbf{y}).$$

4. COMBINING EXACT AND ITERATIVE PEELING TO IMPROVE EFFICIENCY OF THE SAMPLER

When exact peeling is used to sample genotypes, the samples are independent and are drawn from the joint posterior distribution. Thus, there is no need to use the Metropolis-Hastings algorithm. On the other hand, when iterative peeling is used, the samples are not obtained from the joint posterior distribution, and the Metropolis-Hastings algorithm has to be used to accept the proposal. Although the candidate draws obtained by iterative peeling are independent, the Metropolis-Hastings algorithm causes the samples to be dependent when candidates are rejected. The resulting loss in efficiency may be minimized by combining exact and iterative peeling [37]. Exact peeling is used as long as the cutset size is not too large for efficient computations. Then, iterative peeling is used on the remaining part of the pedigree as described below.

To illustrate how exact peeling is combined with iterative peeling, consider the small pedigree shown in Figure 2. We peel individual 5 exactly and apply

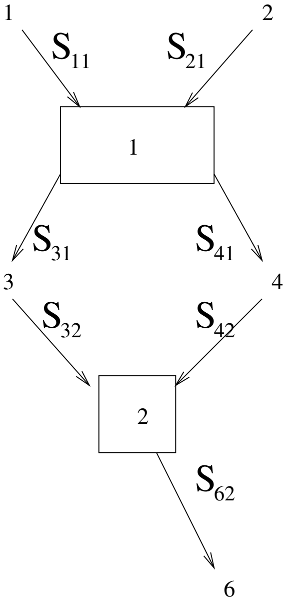


Figure 5. Two-generational pedigree after peeling out individual 5.

iterative peeling to the remaining part of the pedigree. Peeling individual five results in a cutset $c_5(g_3, g_4, g_6)$ of size three. This cutset is represented by a square in Figure 5.

After exact peeling has been used, the graph contains three types of nodes: individual, nuclear-family, and cutset nodes. Further, it contains two types of edges: edges between individuals and nuclear-family nodes and between individuals and cutset nodes.

In iterative peeling, both types of edges need to be updated. The edges between individuals and nuclear-family nodes are updated as described in Section 3. The edges between individuals and cutset nodes are updated as

$$S_{js} = \sum c_s(g_{s_1}, \ldots, g_{s_n}) \prod_{\substack{l \in c_s \\ l \neq j}} R_{sl} \tag{11}$$

where the summation is over the genotypes g_{s_1} to g_{s_n} of the individuals included in cutset c_s, except for the genotype of individual j, and $s_1, \ldots, s_n$ are the individuals in cutset c_s.

After a sufficient number of iterations, we sample genotypes for all "unpeeled" individuals in the pedigree as described in Section 3. Note that in addition to updating the nuclear-family nodes, cutset nodes also need to be updated. To update cutset nodes, (11) is used, but the summation is only over the unsampled genotypes. However, a neighbor of individual j is now defined as any individual who is also a member of any node to which individual j belongs. Once genotypes for all "unpeeled" individuals are sampled, we sample genotypes of the "peeled" individuals in the inverse order of peeling as in Section 2. For example, in Figure 5, iterative peeling is used to sample genotypes of the individuals that were not peeled out exactly (individuals 1, 2, 3, 4, and 6). Once these individuals are sampled, the genotype for 5 is sampled from $f(g_5|\mathbf{y}, g_3, g_4, g_6)$, which is computed as

$$f(g_5|\mathbf{y}, g_3, g_4, g_6) = \frac{f(y_5|g_5)c_5(g_3, g_4, g_6)}{\sum_{g_5} f(y_5|g_5)c_5(g_3, g_4, g_6)}.$$

5. EVALUATION OF THE SAMPLER

One possible approach to computing genotype probabilities is exact peeling. But this approach is extremely inefficient in large pedigrees because peeling must be done for every genotype and for every individual. Furthermore, if the pedigree has large and complex loops, exact peeling cannot be performed. Iterative peeling can be used to approximate the calculations in large pedigrees with complex loops [6,21,36,37]. Thus, approximate probabilities computed by iterative peeling were used to compare with the estimates obtained from different versions of the ESIP sampler. In one of these samplers, exact peeling is applied to the entire pedigree, and samples are obtained directly from the target distribution by reverse peeling. This version is called the *Direct* sampler. Note that, contrary to exact peeling, the *Direct* sampler is very efficient because the entire pedigree is peeled only once, and then genotypes are sampled. However, the *Direct* sampler cannot be used with complex pedigrees, as the cutset sizes become too large for efficient computation. Let m be the size of the largest cutset when exact peeling is applied to the entire pedigree. When m is too large for exact peeling of the entire pedigree, exact peeling is applied until the cutset size is k ($k < m$), and then iterative peeling is applied to the remainder. This is called the ESIP-k sampler. Note that ESIP-m is the *Direct* sampler. For the ESIP samplers and iterative peeling we used five iterations to update edges. Results from the scalar-Gibbs sampler were also obtained for comparison.

After convergence is reached, the different versions of ESIP yield samples from the target distribution. However, the *Direct* sampler does not require a burn-in period, and its samples are independent; thus the effective chain size [10] is equal to the actual sample size. For ESIP-k samplers, as k approaches m, the required burn-in period approaches zero, and the effective chain size approaches the actual chain size. The ESIP-2 sampler is expected to require the longest burn-in period and have the smallest effective chain size.

Table II. Ranges, means, and standard deviations (S.D.) of the absolute differences between genotype probabilities obtained by the PAP program and iterative peeling.

	Range	Mean	S.D.
$P(AA)$	0 to 3.2×10^{-2}	1.0×10^{-2}	6.7×10^{-3}
$P(Aa)$	0 to 5.1×10^{-2}	2.3×10^{-2}	1.7×10^{-3}
$P(aa)$	0 to 4.2×10^{-2}	1.4×10^{-2}	1.3×10^{-3}

5.1. Assessing performance of algorithm using a hypothetical pedigree

To assess the performance of the algorithm, we considered inheritance at a single biallelic disease locus in a hypothetical pedigree with loops. The pedigree is shown in Figure 6. This pedigree consists of two affected and 75 unaffected individuals from four generations. Further, each nuclear family has five or more offspring. The assumed gene frequencies were 0.75 for the good allele A and 0.25 for the bad allele a. Genotypes were sampled for the 75 individuals with missing genotypes using the ESIP-2, ESIP-3, and ESIP-4 = *Direct* samplers. Genotype probabilities were estimated from the samples. Genotype probabilities were also estimated using the scalar-Gibbs sampler.

In this small pedigree we can compute genotype probabilities by exact peeling. These exact calculations were verified with the results from the Package for Pedigree Analyses (PAP) [15]. The probabilities obtained by PAP are considered as the true results. The absolute differences between probabilities obtained by our algorithm using exact peeling and those from PAP are at most 4.9×10^{-5}. These small differences are due to rounding errors.

Approximate genotype probabilities were also computed by iterative peeling [6,21,36,37] and are compared with PAP results (Tab. II). For this pedigree, iterative peeling seems to provide a fairly good approximation.

Results from PAP were also compared with estimates from the ESIP-2, ESIP-3, ESIP-4 = *Direct*, and the scalar-Gibbs samplers (Tab. III). The length of the chain was 10 000 including a burn-in period of 5 000 for the ESIP-2, ESIP-3, and one of the scalar-Gibbs samplers. Thus the genotype probabilities were estimated from the second half of the chain. The length of the chain for the *Direct* sampler was 5 000 with no burn-in period.

Probabilities obtained from ESIP-2, ESIP-3, and *Direct* samplers are close to those obtained by PAP. However, the differences between PAP and the ESIP samplers are larger than 4.9×10^{-5}. This is because in addition to rounding differences, probabilities estimated from samples contain sampling errors. The differences due to sampling can be reduced by increasing the length of the chain.

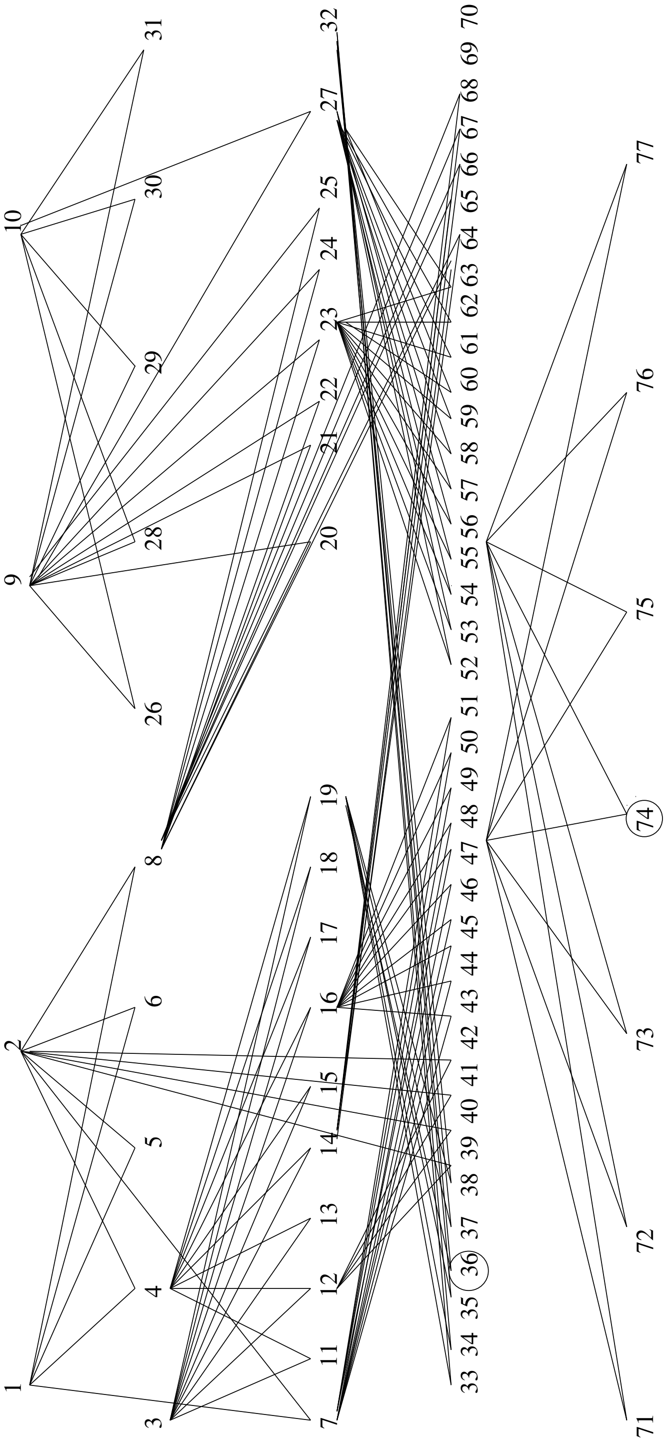


Figure 6. Hypothetical pedigree of 77 individuals. Individuals 36 and 74 are affected.

Table III. Ranges, means, and standard deviations (S.D.) of the absolute differences between genotype probabilities obtained by the PAP program and the ESIP-2, ESIP-3, ESIP-4, and scalar-Gibbs samplers.

Comparison	Range	Mean	S.D.
PAP–ESIP-2 (1)			
$P(AA)$	0 to 2.0×10^{-2}	7.0×10^{-3}	5.2×10^{-3}
$P(Aa)$	0 to 2.9×10^{-2}	8.2×10^{-3}	6.9×10^{-3}
$P(aa)$	0 to 3.0×10^{-2}	7.0×10^{-3}	7.0×10^{-3}
PAP–ESIP-3 (1)			
$P(AA)$	0 to 2.0×10^{-2}	3.9×10^{-3}	3.9×10^{-3}
$P(Aa)$	0 to 1.9×10^{-2}	5.6×10^{-3}	3.9×10^{-3}
$P(aa)$	0 to 1.6×10^{-2}	5.6×10^{-3}	3.8×10^{-3}
PAP–ESIP-4 (2)			
$P(AA)$	0 to 1.3×10^{-2}	3.9×10^{-3}	3.2×10^{-3}
$P(Aa)$	0 to 2.3×10^{-2}	6.9×10^{-3}	4.9×10^{-3}
$P(aa)$	0 to 1.7×10^{-2}	5.3×10^{-3}	3.7×10^{-3}
PAP–Scalar-Gibbs (1)			
$P(AA)$	2.0×10^{-4} to 0.385	0.1159753	0.0936219
$P(Aa)$	1.7×10^{-3} to 0.714	0.1225597	0.1259900
$P(aa)$	2.0×10^{-4} to 0.801	0.1657766	0.1640349
PAP–Scalar-Gibbs (3)			
$P(AA)$	0 to 7.9×10^{-2}	2.2×10^{-2}	1.7×10^{-2}
$P(Aa)$	0 to 7.2×10^{-2}	2.0×10^{-2}	1.7×10^{-2}
$P(aa)$	0 to 0.1011140	2.2×10^{-2}	1.6×10^{-2}

(1) Chain length = 10 000 including a burn-in period of 5 000.
(2) Chain length = 5 000 with no burn-in period.
(3) Chain length = 235 000 including a burn-in period of 200 000.

In contrast to the probabilities obtained from the ESIP samplers, those from a scalar-Gibbs sampler of the same length are very different from the PAP results (Tab. III). For the ESIP samplers, the mean difference with PAP probabilities is 5.5×10^{-3}, and the largest difference is 2.5×10^{-2}. For scalar-Gibbs, however, the mean difference with PAP probabilities is 0.13, and the largest difference is 0.8.

Genotype probabilities were also obtained using the scalar-Gibbs sampler with longer chains. Even with a chain of 235 000 including a burn-in period of 200 000, the differences with PAP probabilities are larger than those for the ESIP samplers (Tab. III).

Table IV. Effective Chain Size for three versions of the ESIP and scalar-Gibbs samplers.

Indiv.	ESIP-2	ESIP-3	*Direct*	Scalar-Gibbs
1	11 380	22 617	24 507	129
2	11 222	21 579	24 008	71
3	11 522	21 717	24 678	64
4	11 465	21 891	24 697	209
5	11 796	21 987	24 067	117
6	11 791	22 354	24 532	582
7	12 150	22 606	25 233	227
8	13 597	22 716	24 148	89
9	10 524	22 741	24 819	385
10	11 059	22 683	23 274	129
11	8 313	22 728	24 857	671

We also compare the performance of the ESIP sampler to scalar-Gibbs by estimating the effective chain size (ECS) [10]. ECS is the size of a chain with independent elements that has the same information content as the actual chain.

To estimate ECS, a chain length of 125 000 was obtained for each sampler. ECS was estimated for 11 individuals chosen at random using the last 25 000 elements of the chain, *i.e.*, ECS was calculated using the elements of the chain after burn-in. The results are shown in Table IV.

Here, we observe that there is a large difference in ECS for the ESIP samplers and scalar-Gibbs; the ESIP samplers result in larger ECS values than the Gibbs sampler. This shows that the Gibbs chain is more correlated. Among the three versions of the ESIP sampler, the *Direct* sampler has the largest ECS values. This is expected, because with the *Direct* sampler, elements in the chain are independent. Thus, the ECS value for the *Direct* sampler should be equal to the size of the chain used for estimation. The observed difference is due to sampling. ECS values for the ESIP-3 sampler are lower than but quite close to those for the *Direct* sampler. This indicates that the proposal distribution in ESIP-3 is a good approximation of the target distribution. ECS values for the ESIP-2 sampler are much lower than those for the *Direct* and ESIP-3 samplers. This indicates that the proposal distribution in ESIP-2 is not as close to the target distribution as it is in ESIP-3.

5.2. Application of the algorithm to a real pedigree

A pedigree that consists of 3 223 dogs (Labrador Retrievers) from "The Seeing Eye, Inc." was used to test the algorithm. The trait of interest for this pedigree is a disease called progressive retinal atrophy (PRA). This disease is transmitted by a recessive allele, and the dog is affected when it has the recessive homozygous genotype. The gene frequencies reported by "The Seeing Eye, Inc." were 0.75 for the good allele *A*, and 0.25 for the bad allele *a*. Of the 3 223 dogs in the pedigree, the eyes of 1 114 dogs have been tested either by an electro-retinalgram (ERG) exam at 18 months of age or older, or by an ophthalmic exam at five years of age or older. Among the 1 114 tested dogs, 35 have the disease, and thus these 35 dogs are known to have the recessive homozygous genotype. For the remaining 1 079 dogs that were tested and found not affected, the genotype could be homozygous dominant (noncarriers) or heterozygous (carriers). The 2 109 dogs that were not tested could have any of the three genotypes. Thus, it is of interest to estimate genotype probabilities to identify dogs that have a high risk of transmitting the PRA gene to their offspring, *i.e.*, dogs that have a high probability of being either heterozygous or homozygous recessive.

Exact peeling methods cannot be used for this pedigree because it has 782 loops, and the size of the largest cutset is 27 when the peeling order was determined as described in Section 2. Thus, the ESIP-7 sampler with a chain length of 125 000 including burn-in period of 100 000 is used as the standard for comparisons. The results from this sampler are compared with those from ESIP-2, ESIP-5, and from scalar-Gibbs samplers using the same chain length and burn-in period as in the ESIP-7 sampler above. Further, to examine the effect of chain length and burn-in period, genotype probabilities were estimated using ESIP-7 with a chain length of 25 000 and with no burn-in period (ESIP-7*). Finally, approximate probabilities were also obtained by iterative peeling.

It is well known that the scalar-Gibbs sampler requires an initial genotypic configuration that is consistent with the observed data. To obtain an initial sample for a founder, the genotype was sampled conditional on its phenotype, and for a nonfounder, the genotype was sampled conditional on its parent genotypes and its phenotype. However, this often resulted in inconsistent samples, because the parents are not sampled conditional on their offspring, and thus some unaffected parents of affected offspring were not sampled as carriers. Thus, the strategy to obtain an initial sample was modified by assigning a heterozygous genotype to these parents.

When the chain length was 125 000, the rejection rates for ESIP-2, ESIP-5, and ESIP-7 were 55.85%, 29.43%, and 22.84%, respectively. In ESIP-7 more individuals are peeled out exactly, and therefore the proposal is closer to the target distribution. This explains why the rejection rate is lower for the ESIP-7 sampler. The computing times, for a Pentium Pro 200, were 36 h 43 min, 42 h 13 min, and 55 h 40 min for ESIP-2, ESIP-5, and ESIP-7, respectively. For each genotype, the range, mean, and standard deviation of the absolute differences of genotype probabilities between ESIP-7 and ESIP-2, ESIP-5, and iterative peeling are given in Table V.

Table V. Ranges, means, and standard deviations (S.D.) of the absolute differences between probabilities obtained by ESIP-7 and ESIP-2, ESIP-5, and iterative peeling.

Comparison	Range	Mean	S.D.
ESIP-7 $^{(1)}$–ESIP-2 $^{(1)}$			
$P(AA)$	0 to 3.7×10^{-2}	6.6×10^{-3}	5.7×10^{-3}
$P(Aa)$	0 to 3.7×10^{-2}	6.8×10^{-3}	5.8×10^{-3}
$P(aa)$	0 to 2.4×10^{-2}	1.3×10^{-3}	2.6×10^{-3}
ESIP-7 $^{(1)}$–ESIP-5 $^{(1)}$			
$P(AA)$	0 to 2.0×10^{-2}	4.1×10^{-3}	3.5×10^{-3}
$P(Aa)$	0 to 2.2×10^{-2}	4.1×10^{-3}	3.5×10^{-3}
$P(aa)$	0 to 1.4×10^{-2}	9.1×10^{-4}	1.7×10^{-3}
ESIP-7 $^{(1)}$–Iterative peeling			
$P(AA)$	0 to 7.9×10^{-2}	6.2×10^{-3}	9.5×10^{-3}
$P(Aa)$	0 to 8.0×10^{-2}	5.8×10^{-3}	8.7×10^{-3}
$P(aa)$	0 to 4.7×10^{-2}	1.6×10^{-3}	3.9×10^{-3}
ESIP-7 $^{(1)}$–ESIP-7* $^{(2)}$			
$P(AA)$	0 to 2.2×10^{-2}	4.0×10^{-3}	3.5×10^{-3}
$P(Aa)$	0 to 2.2×10^{-2}	4.1×10^{-3}	3.5×10^{-3}
$P(aa)$	0 to 1.5×10^{-2}	8.9×10^{-4}	1.6×10^{-3}
ESIP-7 $^{(1)}$–Scalar-Gibbs $^{(1)}$			
$P(AA)$	0 to 1	0.2569	0.2218
$P(Aa)$	0 to 1	0.2061	0.1885
$P(aa)$	0 to 1	0.0663	0.1072

(1) Chain length = 125 000 including a burn-in period of 100 000.
(2) Chain length = 25 000 with no burn-in period.

The largest absolute difference in Table V between the ESIP samplers is 3.7×10^{-2}, thus it is clear that the three ESIP samplers gave similar probabilities. It is more efficient time-wise to use the ESIP-2 sampler, but the rejection rate is almost two times larger than the rejection rate in ESIP-7.

In Table V, the approximate probabilities obtained by iterative peeling are also compared with those estimated using ESIP-7. Here, the mean absolute difference is 0.004, and the largest absolute difference is 0.08. The computing time for iterative peeling was 85 s. To further examine the accuracy of the iterative peeling probabilities, a histogram of the absolute differences between these probabilities and those obtained using ESIP-7 is presented in Figure 7.

The histogram shows that the vast majority of the absolute differences are between 0 and 0.02. These results indicate that iterative peeling is a very good approximation for marginal genotype probabilities.

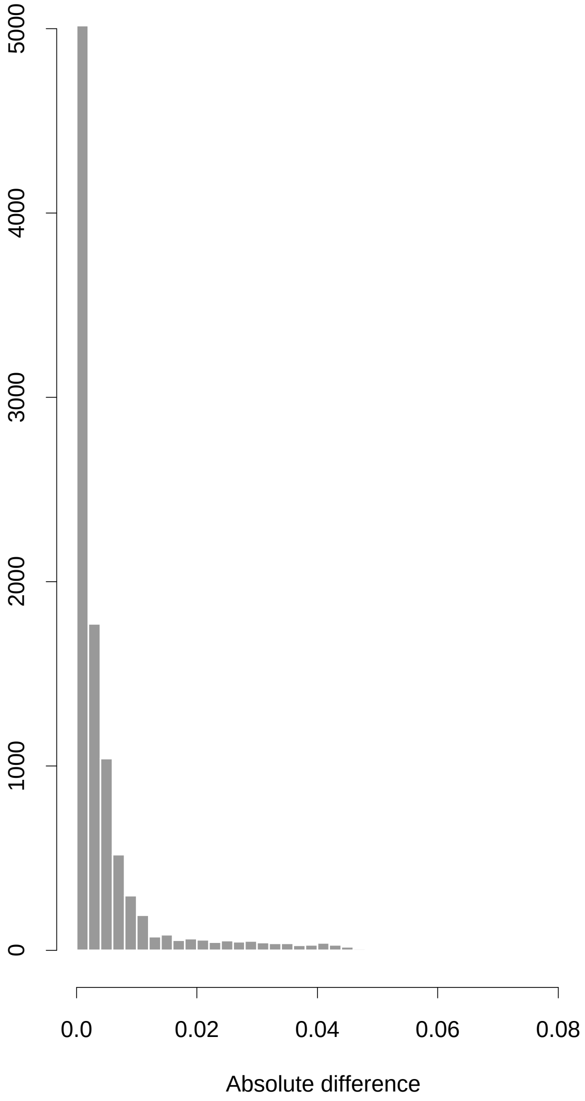


Figure 7. Histogram of the absolute differences between probabilities obtained using the ESIP-7 sampler and iterative peeling.

The probabilities estimated by the two versions of the ESIP-7 sampler are compared in Table V. The means and standard deviations of the absolute differences in this table are very similar to those between ESIP-7 and ESIP-5, both with a chain length of 125 000. This shows that the ESIP-7* sampler is as close to ESIP-7 as the ESIP-5 sampler is to ESIP-7. However, the computation time for the ESIP-7* sampler was 9 h 13 min.

Table VI. Effective chain size for different versions of the ESIP sampler and scalar-Gibbs sampler.

Indiv.	ESIP-2	ESIP-5	ESIP-7	ESIP-8	ESIP-9	Scalar-Gibbs
1	2 645	10 048	10 571	13 918	12 718	24 548
2	4 319	10 422	13 035	13 434	13 213	24 525
3	3 068	9 954	14 417	13 882	13 185	24 915
4	3 554	9 367	12 380	14 218	11 358	24 842
5	1 447	5 882	7 379	11 722	12 336	24 647
6	4 640	12 129	13 667	14 172	14 114	24 518

Finally, in Table V, the probabilities estimated by scalar-Gibbs are compared to those estimated using ESIP-7. For all three genotypes, the smallest values of the ranges were zero. These result from the probabilities of the 35 individuals that were affected and for which the genotype could be determined without error from the phenotype. Moreover, for each genotype, the largest values of the ranges were one. This is because for some individuals, ESIP-7 and scalar-Gibbs get an estimate of probability one for different genotypes. The mean absolute difference between probabilities obtained by scalar-Gibbs and ESIP-7 is 0.156. This is about 40 times larger than the mean absolute difference between probabilities obtained by iterative peeling and the ESIP-7 sampler.

A histogram of these differences is presented in Figure 8. It is clear that the results obtained by the scalar-Gibbs sampler are very different from those obtained by the ESIP-7, which gave results that are in very good agreement with those from the other samplers and iterative peeling. This lack of agreement between the results from the scalar-Gibbs sampler and those from the other approaches may be due to failure of the scalar-Gibbs to converge or due to slow mixing after convergence. To examine if mixing was slow, the effective chain size (ECS) was computed for six individuals using the last 25 000 elements of the chain for the scalar-Gibbs sampler and the first 25 000 elements of the chain for the ESIP-*k* samplers (Tab. VI).

ECS was much larger for scalar-Gibbs than for the ESIP-*k* samplers. Although this seems to indicate that mixing was not a problem, the genotype probabilities estimated from the sample of genotypes obtained by scalar-Gibbs greatly differ from those obtained by iterative peeling and the ESIP sampler. This indicates that the sample obtained from scalar-Gibbs was not representative of the posterior distribution of the genotypes. This shows that it is possible to have a large value for ECS without the chain yielding a representative sample from the target distribution. The above can happen when the chain moves freely among the sampled genotypes but stays within a local area of the target distribution. It is easy to construct a pedigree where the above can be observed. For example, consider the following pedigree consisting of one

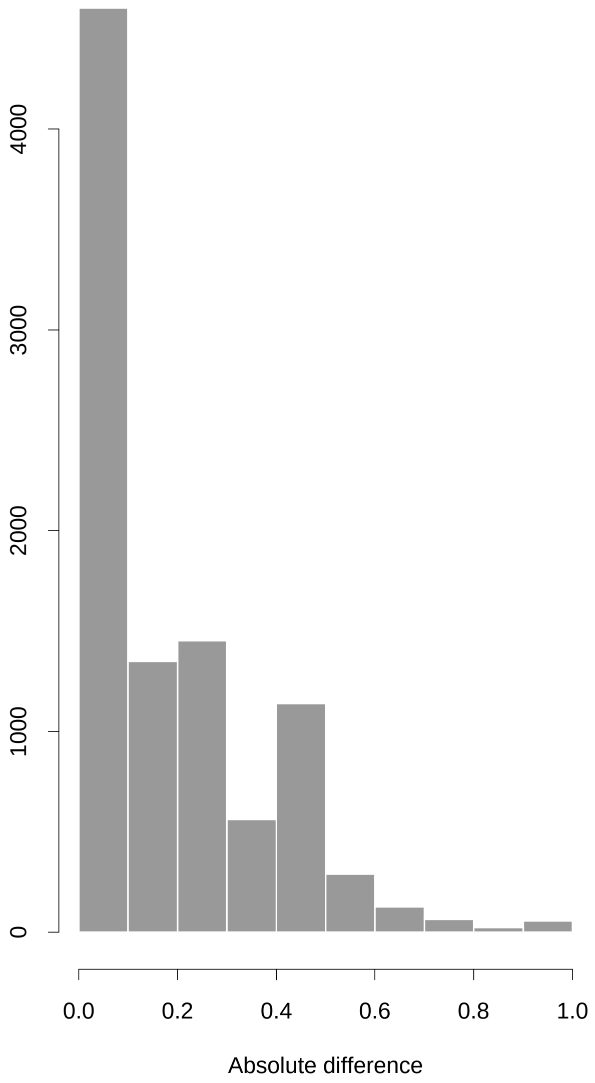


Figure 8. Histogram of the absolute differences between probabilities obtained using the ESIP-7 and scalar-Gibbs samplers.

big nuclear family with 35 offspring. The genotype for 34 of the offspring is known: 17 are heterozygous (*Aa*) and 17 are homozygous (*AA*). The genotype for the parents and one offspring is unknown. It is assumed that the frequency of allele *A* is 0.75. Scalar-Gibbs and ESIP samplers were used to sample the missing genotypes. ECS was computed based on a chain length of 10 000 for both samplers. For the ESIP sampler, ECS values were 9 893 and 9 934 for the parents, and 9 934 for the offspring. For scalar-Gibbs, ECS values

Table VII. Marginal posterior probabilities of missing genotypes in a large nuclear family. Genotype probabilities were calculated exactly by peeling and estimated from 10 000 samples obtained by ESIP and scalar-Gibbs.

Individual	Method	*AA*	*Aa*	*aa*
Parent 1	Exact	0.5	0.5	0
	ESIP	0.4931	0.5069	0
	Scalar-Gibbs	1	0	0
Parent 2	Exact	0.5	0.5	0
	ESIP	0.5069	0.4931	0
	Scalar-Gibbs	0	1	0
Offspring	Exact	0.5	0.5	0
	ESIP	0.4957	0.5043	0
	Scalar-Gibbs	0.4856	0.5144	0

were not defined for the parents, because in all samples the genotypes for the parents were *AA* and *Aa*. However, the ECS value for the offspring was 9 998. The marginal posterior genotype probabilities for the three individuals with missing genotypes are presented in Table VII. From this table it is clear that the scalar-Gibbs sampler did not yield a representative sample from the target distribution, even though the ECS was large for the offspring.

6. COMPUTING TIME OF THE ESIP SAMPLER

The computing time of the ESIP sampler can be split into two components: the time involved in peeling and the time involved in sampling. Peeling time increases exponentially with cutset size *k*, but because peeling is done only once, for small values of *k*, the time for peeling is negligible compared with the time for obtaining many samples. As explained below, sampling genotypes of individuals that were iteratively peeled may be more time consuming than sampling genotypes of individuals that were exactly peeled. Before sampling genotypes of an individual that was iteratively peeled, all its edges must be updated to reflect the already sampled individuals. Some of these edges may be between the individual and cutset nodes of high dimension. Updating these edges can be very time consuming if only a few individuals in the cutset have been sampled. This updating step is not present when sampling genotypes of individuals that were peeled exactly.

To illustrate these concepts, computing times were recorded. The computing times for the hypothetical pedigree were 7, 6 and 1 s for the ESIP-2, ESIP-3, and ESIP-4 samplers, respectively. In all cases the chain length was 2 000. Thus, it is evident that the computing time is minimum when the entire pedigree is exactly peeled (ESIP-4 = *Direct* sampler), and therefore sampling is efficient.

Table VIII. Exact peeling and sampling times for the dog pedigree using different cutset sizes (chain length = 100).

Cutset size	Peeling time (s)	Sampling time (s)	Total
ESIP-2	3	36	39
ESIP-3	3	36	39
ESIP-4	3	33	36
ESIP-5	4	34	38
ESIP-6	5	38	43
ESIP-7	9	48	57
ESIP-8	34	106	140
ESIP-9	70	203	273

Computing times for the dog pedigree using different cutset sizes are presented in Table VIII. The chain length in this case was 100.

Table VIII shows that computing times do not differ between the ESIP-2 sampler and the ESIP-7 sampler, but for $k > 7$ the computing time increases rapidly. With $k = 9$, 59 individuals were peeled iteratively. In sampling the genotypes of these 59 individuals some edges connected to cutsets of high dimension are updated. Therefore, sampling genotypes of these individuals is time consuming. For this pedigree, if m had been nine, the computing time would have been dramatically reduced.

7. SUMMARY AND CONCLUSIONS

The scalar-Gibbs sampler is known to have slow mixing when the pedigree contains large progeny groups, and it may not be irreducible when sampling genotypes at marker loci with more than two alleles [26,30]. Blocking Gibbs has been proposed to solve the problem of slow mixing and reduce the problem of reducibility [20]. As will be shown in a subsequent paper, the ESIP sampler is guaranteed to produce an irreducible chain. This paper gives a detailed description of this sampler.

A small hypothetical pedigree was used to validate the ESIP-k sampler. For this pedigree, exact probabilities were obtained by peeling and compared with those estimated by the ESIP-k samplers. The comparisons indicate that probabilities estimated by the ESIP-k samplers ($k = 2, 3, 4$) using a chain length of 10 000, including a burn-in of 5 000, were accurate. Besides, we observe that increasing cutset size k dramatically increased the ECS. For this small pedigree the computing time was optimal for the ESIP-4 = *Direct* sampler. Thus, considering both ECS and computing time, the most efficient sampler was the *Direct* sampler, *i.e.*, when the samples were obtained directly from the joint posterior distribution.

A real pedigree was also used to test the algorithm. Estimates of the genotype probabilities obtained from different ESIP-*k* samplers were very similar. Furthermore, genotype probabilities computed by iterative peeling were similar to those estimates obtained by the ESIP-*k* samplers, indicating that iterative peeling provides a very good approximation for marginal genotype probabilities. On the other hand, estimates of genotype probabilities obtained by the scalar-Gibbs sampler were very different from those estimated by the ESIP-*k* samplers and probabilities computed by iterative peeling. Thus, we conclude that the scalar-Gibbs sampler failed to converge.

For the ESIP-*k* samplers, as expected, computing time increased exponentially with cutset size and so did ECS. However, for ESIP-*k* samplers with $k \leq 7$ the computing time was about the same. Thus, for the dog pedigree we found that the ESIP-7 sampler was most efficient, because with $k = 7$, the ECS per unit of time was maximum.

In conclusion, the ESIP sampler described in this paper can be used to sample genotypes from complex pedigrees where the scalar-Gibbs sampler has very poor mixing. These samples can be used to estimate genotype probabilities, however, the approximate probabilities from iterative peeling seem to be equally accurate. Furthermore, genotype samples can be used to overcome the computational problems in extended models, where in addition to the effect of the “major” locus, the model includes non-genetic fixed and random effects, and random polygenic effects. In these extended models, in addition to samples of genotypes at the major locus, samples are also needed for the other fixed and random effects in the model. Using the scalar-Gibbs to obtain these samples may also result in poor mixing. Fortunately, García-Cortés and Sorensen [8] have described an efficient method to jointly sample the random and fixed effects in a linear model.

ACKNOWLEDGEMENTS

Journal Paper No. J-19049 of the Iowa Agriculture and Home Economics Experiment Station, Ames, Iowa. Project No. 3146, and supported by Hatch Act and State of Iowa funds.

The authors are grateful to Dr. Eldin A. Leighton, Director of Canine Genetics of “The Seeing Eye, Inc.,” who provided the pedigree data used in this study. The authors would like to thank the referees for pointing out that poor mixing can also result from horizontal dependence between loci.

REFERENCES

[1] Bonney G.E., Compound regressive models for family data, Hum. Hered. 42 (1992) 28–41.

[2] Cannings C., Thompson E.A., Skolnick E., Probability functions on complex pedigrees, Adv. Appl. Prod. 10 (1978) 26–61.

[3] Elston R.C., Stewart J., A general model for the genetic analysis of pedigree data, Hum. Hered. 21 (1971) 523–542.

[4] Fernández S.A., Fernando R.L., Determining peeling order using sparse matrix algorithms, J. Dairy Sci. (Submitted).

[5] Fernández S.A., Fernando R.L., Carriquiry A.L., An algorithm to sample marker genotypes in a pedigree with loops, in: Proceedings of the American Statistical Association, Section on Bayesian Statistical Science, Alexandria, VA, 1999, pp. 60–65.

[6] Fernando R.L., Stricker C., Elston R.C., An efficient algorithm to compute the posterior genotypic distribution for every member of a pedigree without loops, Theor. Appl. Genet. 87 (1993) 89–93.

[7] Fernando R.L., Stricker C., Elston R.C., The finite polygenic mixed model: an alternative formulation for the mixed model of inheritance, Theor. Appl. Genet. 88 (1994) 573–580.

[8] García-Cortés L.A., Sorensen D., On a multivariate implementation of the Gibbs sampler, Genet. Sel. Evol. 28 (1996) 121–126.

[9] Gelman A., Carlin J.B., Stern H.S., Rubin D.B., Bayesian data analysis, Chapman & Hall, London, HS, 1995.

[10] Geyer C., Practical Markov chain Monte Carlo, Stat. Sci. 7 (1992) 473–511.

[11] Gilks W.R., Richardson S., Spiegelhalter D.J., Markov chain Monte Carlo in practice, Chapman & Hall, London, HS, 1996.

[12] Guo S.W., Thompson E.A., A Monte Carlo method for combined segregation and linkage analysis, Am. J. Hum. Genet. 51 (1992) 1111–1126.

[13] Hasstedt S.J., A mixed model approximation for large pedigrees, Comput. Biomed. Res. 15 (1982) 195–307.

[14] Hasstedt S.J., A variance components/major locus likelihood approximation on quantitative data, Genet. Epidemiol. 8 (1991) 113–125.

[15] Hasstedt S.J., Pedigree Analysis Package, revision 4.0 edn. Department of Human Genetics, University of Utah, Salt Lake City, UT, 1994.

[16] Heath S.C., Generating consistent genotypic configurations for multi-allelic loci and large complex pedigrees, Hum. Hered. 48 (1998) 1–11.

[17] Hoeschele I., VanRaden P.M., Bayesian analysis of linkage between genetic markers and quantitative trait loci. I. Prior knowledge, Theor. Appl. Genet. 85 (1993a) 953–960.

[18] Hoeschele I., VanRaden P.M., Bayesian analysis of linkage between genetic markers and quantitative trait loci. II. Combining prior knowledge with experimental evidence, Theor. Appl. Genet. 85 (1993b) 946–952.

[19] Hoeschele P., Uimari P., Grignola F.E., Zhang Q., Gage K.M., Advances in statistical methods to map quantitative trait loci in outbred populations, Genetics 147 (1997) 1445–1457.

[20] Janss L.L.G., Thompson R., van Arendonk J.A.M., Application of Gibbs sampling for inference in a mixed major gene-polygenic inheritance model in animal populations, Theor. Appl. Genet. 91 (1995) 1137–1147.

[21] Janss L.L.G., van Arendonk J.A.M., van der Werf J.H.J., Computing approximate monogenic model likelihoods in large pedigrees with loops, Genet. Sel. Evol. 27 (1995) 567–579.

[22] Jensen C.S., Kong A., Blocking Gibbs sampling for linkage analysis in large pedigrees with many loops, Am. J. Hum. Genet. 65 (1999) 885–901.

[23] Jensen C.S., Kong A., Kjærulff U., Blocking Gibbs sampling in very large probabilistic expert systems, Int. J. Hum. Comp. Stud. 42 (1995) 647–66.

[24] Lange K., Boehnke M., Extensions to pedigree analysis. V. Optimal calculation of Mendelian likelihoods, Hum. Hered. 33 (1983) 291–301.

[25] Lange K., Elston R.C., Extensions to pedigree analysis. I. Likelihood calculations for simple and complex pedigrees, Hum. Hered. 25 (1975) 95–105.

[26] Sheehan N., Genetic restoration on complex pedigrees, PhD thesis, University of Washington, Seattle, WA, 1990.

[27] Stricker C., Fernando R.L., Eltson R.C., An algorithm to approximate the likelihood for pedigree data with loops by cutting, Theor. Appl. Genet. 91 (1995) 1054–1063.

[28] Thomas A., Approximate computations of probability functions for pedigree analysis, IMA J. Math. Appl. Med. Biol. 3 (1986a) 157–166.

[29] Thomas A., Optimal computations of probability functions for pedigree analysis, IMJ J. Math. Appl. Med. Biol. 3 (1986b) 167–178.

[30] Thomas D., Cortessis V., A Gibbs sampling approach to linkage analysis, Hum. Hered. 42 (1992) 63–76.

[31] Thompson E.A., Monte Carlo likelihood in genetic mapping, Stat. Sci. 9 (1994a) 355–366.

[32] Thompson E.A., Monte Carlo likelihood in the genetic mapping of complex traits, Phil. Trans. R. Soc. Lond. B. 344 (1994b) 345–351.

[33] Thompson E.A., MCMC estimation of multi-locus genome sharing and multipoint gene location scores, Intern. Stat. Rev. 68 (2000) 53–73.

[34] Thompson E.A., Heath S.C., Estimation of conditional multilocus gene identity among relatives, in: Seillier-Moseiwitch F., (Ed.), Statistics in molecular biology and genetics: selected proceedings of a 1997 joint AMS-IMS-SIAM Summer conference on statistics in molecular biology, IMS Lect. Note-Monograph Ser. 33, Hayward, CA, 1999, pp. 95–113.

[35] Uimari P., Thaller G., Hoeschele I., The use of multiple markers in Bayesian methods for mapping quantitative trait loci, Genetics 143 (1996) 1831–1842.

[36] Van Arendonk J.A.M., Smith C., Kennedy B.W., Method to estimate genotype probabilities at individual loci in farm livestock, Theor. Appl. Genet. 78 (1989) 735–740.

[37] Wang T., Fernando R.L., Stricker C., Elston R.C., An approximation to the likelihood for a pedigree with loops, Theor. Appl. Genet. 93 (1996) 1299–1309.