

An algorithm to sample unobservable genotypes in complex pedigrees

SOLEDAD A. FERNANDEZ, ROHAN L. FERNANDO and ALICIA L. CARRIQUIRY
Iowa State University, USA

SUMMARY

Probability functions such as likelihoods and genotype probabilities play an important role in the analysis of genetic data. When genotype data are incomplete Markov chain Monte Carlo (MCMC) methods, such as the Gibbs sampler, can be used to sample genotypes at the marker and trait loci. The Markov chain that corresponds to the scalar Gibbs sampler may not work due to slow mixing. Further, the Gibbs chain may not be irreducible when sampling genotypes at marker loci with more than two alleles. These problems do not arise if the genotypes are sampled jointly from the entire pedigree. When the pedigree does not have loops, a joint sample of the genotypes can be obtained efficiently *via* modification of the Elston-Stewart algorithm. When the pedigree has many loops, obtaining a joint sample can be time consuming. We propose a method for sampling genotypes from a pedigree so modified as to make joint sampling efficient. These samples, obtained from the modified pedigree, are used as candidate draws in the Metropolis-Hastings algorithm.

Keywords: METROPOLIS-HASTINGS; PEELING; ITERATIVE PEELING.

1. INTRODUCTION

Determining genotype probabilities is important in genetic counseling, linkage analysis and in genetic evaluation programs. In genetic counseling, for example, it is important to know which individuals in a population are probable carriers of a recessive disease allele. The first methods for determining genotype probabilities were developed in human genetics by Elston and Stewart (1971) and were reviewed by Elston and Rao (1978) and by Elston (1987). Also, several human genetics computer packages are available to compute genotype probabilities. In livestock, pedigrees are usually much larger because animals, specially males, have multiple mates. Thus, the application of computer intensive methods developed for humans will often be difficult or inappropriate in livestock data.

To obtain estimates of the genotype probabilities, the likelihood of the pedigree is needed. The likelihood L is obtained as $L \propto \sum_{\mathbf{g}} f(\mathbf{y}|\mathbf{g})P(\mathbf{g})$, where \mathbf{y} is the vector of phenotypes and \mathbf{g} is the vector of genotypes, $f(\mathbf{y}|\mathbf{g})$ are the conditional probabilities of \mathbf{y} given \mathbf{g} , and $P(\mathbf{g})$ are the genotype probabilities. The summation is over all possible genotypes for all the individuals in the pedigree. The computations involved in the likelihood are not feasible except in trivial examples. For example, assume that there are

two possible alleles for a locus, resulting in 3 possible genotypes for every individual in the pedigree (AA , Aa or aA and aa). If the pedigree consists of 100 individuals then 3^{100} summations need to be performed in order to compute the likelihood as before.

For pedigrees without loops, the likelihood can be computed efficiently using the Elston-Stewart algorithm (Elston and Stewart, 1971), which is also called *peeling*. Generalizations of this algorithm (Cannings et al., 1978; Lange et al., 1975, 1983) provide strategies to compute the likelihood efficiently for general pedigrees with simple loops.

When a mixed model of inheritance is used, the likelihood is not easy to obtain. Under this model, the phenotypic values of individuals in the pedigree cannot be assumed to be conditionally independent given the pedigree members because they are also influenced by the polygenic loci. Markov chain Monte Carlo (MCMC) methods, such as the Gibbs sampler, have been proposed to overcome these problems.

When using the Gibbs sampler, however, mixing can be very slow due to the dependence between genotypes of parents and progeny (Janss et al., 1995). The larger the progeny groups, the stronger the dependence; thus the Gibbs chains do not move. When this happens it is said that the chains are reducible “in practice”.

In this paper we propose a method to sample genotypes from large and complex pedigrees, and apply the proposed algorithm to estimate the genotype probabilities in a Labrador Retriever pedigree under study. We use the Metropolis-Hastings algorithm and sample genotypes (candidates) jointly from a proposal distribution that is close to the true posterior distribution of interest. When there are no loops, our proposal is the “true” posterior distribution and the Metropolis-Hastings algorithm reduces to direct sampling. When there are loops in the pedigree, we construct our proposal distribution by first sampling genotypes using iterative peeling. A variation of this approach consists in peeling the pedigree exactly up to the point where the complexity of loops makes it difficult to continue, and then using Metropolis-Hastings to sample from a “partially” peeled pedigree.

2. THE PEELING APPROACH FOR SAMPLING GENOTYPES

Before describing iterative peeling we discuss the Elston-Stewart algorithm (Elston and Stewart, 1971) known as *peeling*.

Consider the simple pedigree shown in Figure 1. To introduce the principles involved in peeling we show how to sample genotypes from $f(\mathbf{g}|\mathbf{y})$ for a monogenic trait in pedigrees without loops.

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

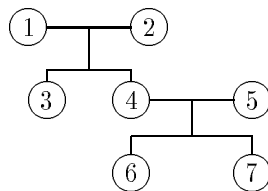


Figure 1. Simple two-generational pedigree.

described below. Thus, 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 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 (Elston and Stewart, 1971; Cannings et al., 1978). The first step in computing $f(g_1|\mathbf{y})$ is to compute the likelihood.

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

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

where $h(g_j) = P(g_j)f(y_j|g_j)$ 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 , $P(g_j|g_m, g_f)$, is the probability that an individual has genotype g_j given parental genotypes g_m and g_f (transition probability).

Computing the likelihood as given by (1) is feasible only for small pedigrees. The Elston-Stewart algorithm can be thought of as providing an efficient reordering of the additions and multiplications in computing the likelihood. Thus, L in (1) is rearranged as

$$\sum_{g_1} \sum_{g_2} h(g_1)h(g_2) \sum_{g_3} h(g_1, g_2, g_3) \sum_{g_4} h(g_1, g_2, g_4) \sum_{g_5} h(g_5) \sum_{g_6} h(g_4, g_5, g_6) \times \sum_{g_7} h(g_4, g_5, g_7). \quad (2)$$

Note that (2) is identical in value to (1) but is computationally more efficient. For example, consider the summation over g_7 . In (1) this summation is done over 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 $h(g_4, g_5, g_7)$, which depends only on two other individual genotypes (g_4 and g_5). In (2), the summation over g_7 is done only for all combinations of values of g_4 and g_5 . An expression involving g_5 and g_4 is obtained after summing out g_7 . These expressions must be stored to be used in the final computation of the likelihood. The result, from peeling an individual is called a *cutset*. For example, after peeling g_7 we obtain a cutset of size 2, $c_7(g_4, g_5) = \sum_{g_7} h(g_4, g_5, g_7)$.

To compute L efficiently, the order of peeling is critical. For example, consider peeling g_1 as the first step, the result is a cutset of size 3, $c_1(g_2, g_3, g_4) = \sum_{g_1} h(g_1)h(g_1, g_2, g_3)h(g_1, g_2, g_4)$. The computation of $c_1(g_2, g_3, g_4)$ involves summing over g_1 for all genotype combinations of g_2, g_3 and g_4 . Computing $c_7(g_4, g_5)$ has lower storage and computational requirements than computing $c_1(g_2, g_3, g_4)$.

Thus, to evaluate the likelihood for this pedigree we first need to define the peeling order. The peeling order is determined as follows. First we list all the individuals that need to be peeled and then sort them according to the size of the cutset that is generated if that individual is peeled. We always start peeling the individual with the smallest cutset in each step.

In this case, the peeling order could be: 7, 6, 5, 4, 3, 2 and 1. Once all individuals have been peeled we sample individual's genotypes in the reverse order to which they were peeled (reverse peeling, Heath, 1998). In this example, after peeling individual 1 we compute the marginal probability for 1 as $f(g_1|\mathbf{y}) = P(g_1)f(y_1|g_1)c_1(g_1)/L$.

Once $f(g_1|\mathbf{y})$ has been obtained, we sample g_1 using the inverse cumulative function. Next, we compute

$$f(g_2|g_1, \mathbf{y}) = f(y_2|g_2)c_4(g_1, g_2)c_3(g_1, g_2) / \left[\sum_{g_2} (y_2|g_2)c_4(g_1, g_2)c_3(g_1, g_2) \right],$$

and then we sample g_2 from $f(g_2|g_1, \mathbf{y})$. Repeatedly applying this 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 $f(g_1|\mathbf{y})$, g_2 from $f(g_2|\mathbf{y}, g_1)$, g_3 from $f(g_3|\mathbf{y}, g_1, g_2)$ and so on.

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

3. ITERATIVE PEELING TO SAMPLE GENOTYPES

Peeling methods cannot be applied when pedigrees are large and have complex loops. Iterative peeling (Van Arendonk et al., 1989; Janss et al., 1992; Wang et al., 1996), however can be used to get approximate results. To describe iterative peeling, it is convenient to present the pedigree as a directed graph (Figure 2 (a)). Before peeling, the graph contains

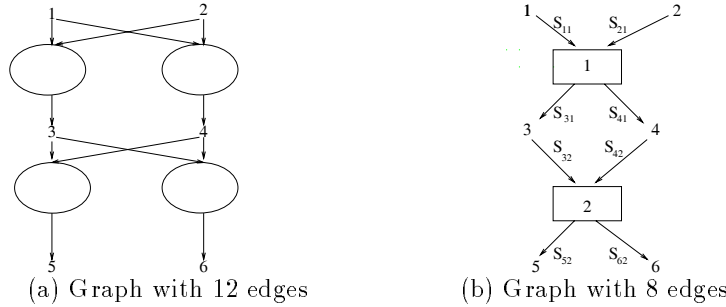


Figure 2. Graph representation of a two-generational pedigree with loops.

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 mating nodes is shown in Figure 2 (b). Here, the nuclear-family nodes are represented by rectangles. There are 8 edges: $S_{11}, S_{21}, S_{31}, S_{32}, S_{41}, S_{42}, S_{52}, S_{62}$ in this graph. The first subindex of S indicates the individual number, and the second subindex indicates the nuclear-family node number. The edges S_{ij} can be interpreted as either conditional or joint probabilities, depending on their location in the graph. We use this small example to explain iterative peeling. The general expressions and more details on the computations are presented in Fernandez et al., (2000).

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 can be obtained from the normalized values of $f(y_1|g_1)P(g_1)S_{11}$. Below we describe how to

iteratively compute S_{11} .

We first initialize all the edge probabilities. Typically, all edge probabilities are initialized to 1. For this example, however it is convenient to set S_{41} to be 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 $f(y_2, y_3, y_4|g_1)$. 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

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

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

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

which is the conditional probability $f(y_4, y_5, y_6|g_3)$. Note that in these three cases, when we multiplied by an edge probability we used the initial values.

Next, we update S_{41} as

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

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 3 (a). Next, we update S_{42} as

$$S_{42} = \sum_{g_1} \sum_{g_2} \sum_{g_5} \sum_{g_6} f(y_5|g_5)P(g_5|g_3, g_4)f(y_6|g_6)P(g_6|g_3, g_4)f(y_3|g_3)P(g_3|g_1, g_2) \underbrace{S_{31}}_{f(y_1, y_2, y_4, g_3)}.$$

Again, in this case we use an edge probability that was already updated, and thus $S_{42} = f(y_1, y_2, y_3, y_4, y_5, y_6|g_3)$, 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 3 (b), given the genotype of individual 4.

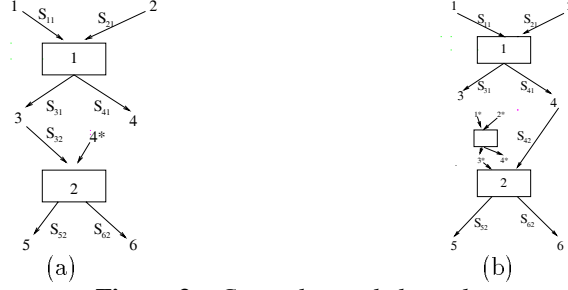


Figure 3. *Cut and extended graphs*

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

$$S_{21} = \sum_{g_3} \sum_{g_4} f(y_1 | g_1) P(g_1) f(y_3 | g_3) P(g_3 | g_1, g_2) f(y_4 | g_4) P(g_4 | g_1, g_2) S_{32} S_{42}, \quad (3)$$

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

$$f(g_2 | \mathbf{y}, g_1) = \frac{f(y_2 | g_2) P(g_2) S_{21}}{\sum_{g_2} f(y_2 | g_2) P(g_2) S_{21}}. \quad (4)$$

This process is continued until all individuals are sampled. We propose to use these sampled genotypes as the proposal distribution in Metropolis-Hastings algorithm to accept or reject the candidate draws.

The general algorithm is the following. First, all edge probabilities are iteratively updated. 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, $f(g_j | \mathbf{y})$. 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 those nuclear-family nodes to which the sampled individual belongs to. To sample a neighbor, we first update all its edges to reflect the already sampled genotypes. To update edges, we use expressions like (3), but the summations are only over the unsampled genotypes. Now to sample the genotype conditional on the already sampled genotypes we use expressions like (4) with the edges that were updated for the sampled genotypes.

3.1 Improving efficiency of sampler

The efficiency of the sampler can be improved by combining exact peeling with iterative peeling. Exact peeling (Section 2.) is used until the size of cutsets gets large enough to make computations unfeasible. Then, iterative peeling is used, which as discussed above is equivalent to cutting and extending the remaining loops.

After iteratively updating all the edge probabilities, we sample genotypes for all individuals in the pedigree. First we sample genotypes for the individuals that were not

peeled out. Once all remaining individuals are sampled, we sample genotypes of the ‘‘peeled’’ individuals in the inverse order of peeling (see Section 2).

3.2 Metropolis-Hastings algorithm

We consider the special case of *independence sampling*. Thus, the acceptance probability (Gilks et al., 1996) is

$$\eta = \min \left(1, \frac{\pi(g_c)q(g_{prev})}{\pi(g_{prev})q(g_c)} \right), \quad (5)$$

where π is the target distribution and 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 chain moves from g_{prev} to g_c with probability η , and it stays at g_{prev} with probability $1 - \eta$. We sample genotypes from the iteratively peeled pedigree (proposal) and use them in the Metropolis-Hastings step to be rejected or accepted. To obtain $\pi(\cdot)$ on the true pedigree, we use $\pi(\mathbf{g}) = \prod_{j=1}^{n_1} P(g_j) \prod_{j=n_1+1}^n P(g_j|g_{f_j}, g_{m_j})$, 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, $\pi(\mathbf{g}) = P(g_1)P(g_2)P(g_3|g_1, g_2)P(g_4|g_1, g_2)P(g_5|g_3, g_4)P(g_6|g_3, g_4)$.

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

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

4. ESTIMATION OF GENOTYPE PROBABILITIES

The proposed sampling method can be used to estimate the genotype probabilities by sampling from a proposal distribution generated by iterative peeling for the entire pedigree or by peeling exactly up to a certain point and then perform iterative peeling.

4.1 Assessing the performance of the algorithm

To assess the performance of the algorithm we used a small pedigree with loops. We considered the inheritance at a single biallelic disease locus. This small pedigree consists of 77 individuals, and two of them are affected. There are four generations in this pedigree and large families (more than five offspring per family). This pedigree also has a few loops. We sampled genotypes for all the individuals and computed the genotype probabilities.

In this small pedigree we can perform exact calculations by exact peeling. These exact calculations were verified with the results from package for pedigree analyses (Hasstedt, 1994; PAP). The probabilities obtained by PAP can be thought as the true results. The range, means and standard deviations of the absolute differences between genotype probabilities from our algorithm and PAP for the 77 individuals are shown in Table 1.

Table 1. Absolute differences between probabilities computed by PAP and exact peeling.

	Range	Mean	St. Dev
$P(AA)$	0 to 4.8×10^{-5}	2.5×10^{-5}	1.6×10^{-5}
$P(Aa)$	0 to 4.9×10^{-5}	2.3×10^{-5}	1.4×10^{-5}
$P(aa)$	0 to 4.9×10^{-5}	2.2×10^{-5}	1.4×10^{-5}

In Table 1 we observe that the genotype probabilities computed by the two methods do not differ. The small differences are due to rounding errors.

We then compare the results from PAP with estimates from the proposed sampling method where no exact peeling was done (Table 2) and also with the estimates from the proposed method where exact peeling was done until the cutset size was 4 (C_4) and then iterative peeling was done for the rest of the pedigree (Table 3). The length of the chain was 10,000 iterations and we discarded the first half, thus the genotype probabilities are obtained based on the second half of the chain. Tables 2 and 3 show the ranges, means and standard deviations for the absolute differences between PAP and the proposed method with no partial peeling and with partial peeling, respectively. We observe that the results are similar, indicating that for this small pedigree there is no advantage in partially peeling the pedigree prior to sampling.

Table 2. *Absolute differences between PAP and proposed method with no exact peeling.*

	Range	Mean	St. Dev
$P(AA)$	0 to 2.1×10^{-2}	5.6×10^{-3}	5.4×10^{-3}
$P(Aa)$	0 to 2.5×10^{-2}	6.9×10^{-3}	5.8×10^{-3}
$P(aa)$	0 to 2.2×10^{-2}	6.3×10^{-3}	5.3×10^{-3}

Table 3. *Absolute differences between PAP and proposed method with exact peeling (C_4).*

	Range	Mean	St. Dev
$P(AA)$	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}

The rejection rates were 29% and 5% for the proposed method with no exact peeling and with exact peeling up to cutset size=4, respectively. Thus, it seems that it is more efficient to peel exactly as much as possible and then perform iterative peeling to the remaining core pedigree.

Results from Tables 2 and 3 show that the proposed method yields accurate estimates. If we increase the number of samples, the estimates improve even more.

4.2 Applications of the methods in a real pedigree

A real dog pedigree was used to test the performance of the sampling algorithm that we just described. The trait of interest in 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 (aa). The pedigree consists of 3,052 dogs (Labrador Retrievers) from ‘‘The Seeing Eye, Inc’’, and 33 of them are known to have the disease. That is, for these 33 dogs we know the genotype and phenotype. For the rest of the dogs we are interested in obtaining estimates of the genotype probabilities to determine

which dogs are at highest risk of transmitting the PRA gene to their offspring and which dogs are at lower risk of both transmitting the gene and of being PRA affected.

Exact peeling methods cannot be used in this pedigree because there are 679 loops that need to be cut. Thus, we used the proposed method. We peeled exactly the pedigree up to cutsets of size 4 (C_4). We then iteratively peeled the remaining core pedigree and sampled genotypes according to Metropolis-Hastings algorithm as described above. The rejection rate was 53% and the length of the chain was 20,000 (first 5,000 were discarded) . This rejection rate is dramatically reduced when more genotypes in the pedigree are known.

We also tried running the program for cutset size=0, that is using the proposal generated by performing iterative peeling for the entire pedigree (with no exact peeling). After three days the program was at iteration 568 and we decided to stop the execution.

In general, it seems that it is more efficient to peel exactly the pedigree as much as possible and then perform iterative peeling to the remaining core pedigree to obtain the proposal distribution to be used in Metropolis-Hastings algorithm. We cannot however, peel too deeply because then the cutsets become large increasing the expense in computation time and memory.

4.3 Results: proportion of dogs carrying the PRA allele

The estimated numbers of affected and carrier animals are presented in Table 4. We observe that the estimated number of affected dogs is 41 (including the 33 dogs known to have the disease).

Table 4. *Estimated number of affected and carrier animals.*

Genotype probability	No. of animals
$0.5 \leq P(aa) < 0.6$	526
$0.6 \leq P(aa) < 0.7$	547
$0.7 \leq P(aa) < 0.8$	284
$0.8 \leq P(aa) < 0.9$	115
$0.9 \leq P(aa) < 1$	161
$P(aa) = 1$	41
$0.5 \leq P(Aa) < 0.6$	482
$0.6 \leq P(Aa) < 0.7$	218
$0.7 \leq P(Aa) < 0.8$	56

5. SUMMARY AND CONCLUSIONS

Estimating the genotype probabilities at biallelic loci is non-trivial when pedigrees are large and contain loops. In this case, a standard scalar Gibbs sampling approach cannot be used, as the chains are reducible in practice. We propose a more general Metropolis-Hastings approach to sampling genotypes jointly from complex pedigrees, in which candidate draws are obtained from modified pedigrees. These modified pedigrees are obtained by applying extensions of traditional peeling methods, and are used as candidate draws in the

Metropolis-Hastings step. The resulting Markov chains satisfy the assumptions that are required for good performance of MCMC methods.

The method for sampling genotypes was developed to address the problem of estimating genotype probabilities in Labrador Retrievers. "The Seeing Eye, Inc." provided the pedigree data of interest that included over 3,000 animals, and had over 600 closed loops created by inbreeding and multiple matings. A summary of the results of this pedigree analysis is given in Table 4.

Another important application of this method is in linkage analyses where the genotypes at the marker loci must be sampled when data are incomplete. The proposed method can be applied to sample genotypes at marker loci with more than two alleles. It can be shown that the resulting chains are irreducible.

REFERENCES

- Cannings, C., Thompson, E.A. and Skolnick, E.H. (1978). Probability functions on complex pedigrees. *Adv. Appl. Prod.* **10**, 26-61.
- Elston, R.C. and Stewart, J. (1971). A general model for the genetic analysis of pedigree data. *Hum Hered* **21**, 523-542.
- Elston, R.C. and Rao, D.C. (1978). Statistical modeling and analysis in human genetics. *Annu. Rev. Biophys. Bioeng.* **7**, 253-286.
- Elston, R.C. (1987). Human quantitative genetics. *Proc. 2nd. Int. Conf. Quant. Genet.*, p.281.
- Fernandez, S.A., Fernando, R.L., Carriquiry, A.L. and Gulbrandtsen, B. (2000). Estimating genotype probabilities in complex pedigrees. *Case Studies in Bayesian Statistics. V*. Berlin: Springer. (In Press).
- Gilks, W.R., Richardson, S. and Spiegelhalter, D.J. (1996). Markov Chain Monte Carlo in Practice. London: Chapman and Hall.
- Hasstedt, S.J. (1994). Pedigree Analysis Package. *Department of Human Genetics, University of Utah*. Revision 4.0.
- Heath, S.C. (1998). Generating consistent genotypic configurations for multi-allelic loci and large complex pedigrees. *Hum. Hered.* **48**, 1-11.
- Janss, L.L.G. and Thompson, R. and Van Arendonk, J.A.M.. (1995) Application of Gibbs sampling for inference in a mixed major gene-polygenic inheritance model in animal populations. *Theor. Appl. Genet.* **91**, 1137-1147.
- Janss, L.L.G. and Van der Werf J.H.J and Van Arendonk J.A.M. (1992). Detection of a major gene using segregation analysis in data from generations. *Proc. Eur. Assoc. Anim. Prod.*.
- Lange, K. and Boehnke, M. (1983). Extensions to pedigree analysis. V. Optimal calculation of Mendelian likelihoods. *Hum. Hered.* **33**, 291-301.
- Lange, K. and Elston, R.C. (1975). Extensions to pedigree analysis. I. Likelihood calculations for simple and complex pedigrees. *Hum. Hered.* **25**, 95-105.
- Van Arendonk, J.A.M. and Smith, C. and Kennedy, B.W. (1989). Method to estimate genotype probabilities at individual loci in farm livestock. *Theor. Appl. Genet.* **78**, 735-740.
- Wang, T. and Fernando, R.L. and Stricker, C. and Elston, R.C. (1996). An approximation to the likelihood for a pedigree with loops. *Theor. Appl. Genet.* **93**, 1299-1309.