T. Wang · R. L. Fernando · C. Stricker · R. C. Elston

An approximation to the likelihood for a pedigree with loops

Received: 7 April 1995 / Accepted: 19 April 1996

Abstract This paper presents a new approximation to the likelihood for a pedigree with loops, based on cutting all loops and extending the pedigree at the cuts. An opimum loop-cutting strategy and an iterative extension technique are presented. The likelihood for a pedigree with loops is then approximated by the conditional likelihood for the entire cut-extended pedigree given the extended part. The approximate likelihoods are compared with the exact likelihoods obtained using the program MENDEL for several small pedigrees with loops. The approximation is efficient for large pedigrees with complex loops in terms of computing speed and memory requirements.

Key words Likelihood · Peeling · Pedigree with loops · Segregation analysis · Linkage analysis

Introduction

The method of maximum likelihood has been used widely in segregation and linkage analyses involving pedigree data. Elston and Stewart (1971) first proposed an efficient algorithm to compute the likelihood for a pedigree without loops. This approach has been extended for pedigrees with loops (Lange and Elston 1975; Cannings et al. 1978; Lange and Boehnke 1983; Thomas 1986 a, b) and used for the analysis of human pedigrees. These methods, however, are not suitable for analysis of livestock pedigrees with complex loops and thousands of individuals. Recently, Monte Carlo procedures, such as Gibbs sampling, have been applied for segregation and linkage analysis (Thompson and Wijsman 1990; Thompson and Guo 1991; Lange and Sobel 1991; Thomas and Cortessis 1992; Guo and Thompson 1994). Gibbs sampling has the potential to overcome the problem of computing the likelihood for large and complex pedigrees. It has been shown, however, that the Gibbs sampler may fail when applied to models with multiple alleles (Lange and Matthysse 1989; Thomas and Cortessis 1992), and methods to overcome this problem have been investigated (Sheehan and Thomas 1993).

Two approaches have been proposed to approximate the likelihood for large and complex livestock pedigrees with loops. The first approach is based on an iterative algorithm (Janss et al. 1992), and the second is based on an algorithm that "cuts" loops (Stricker et al. 1995). The objective of the present paper is to combine these two approaches, together with other significant improvements, to obtain an approximation that is closer to the exact likelihood and that is more efficient than either approach. The approximate likelihood obtained using this new approach is compared with the exact likelihood obtained using the program MENDEL (Lange 1991) for several small pedigrees.

We first present an iterative and a non-iterative algorithms to compute the likelihood for a pedigree without loops. Then we show how these two methods, together with loop-cutting, can be combined to approximate the likelihood for a pedigree with loops.

Definitions

A pedigree is a set of related individuals that includes either both or neither of the two parents of each individual in the pedigree. A pedigree consisting of three families is shown in Fig. 1. This pedigree will be used to illustrate the concepts defined in this section. A phenotypic value is usually associated with each mem-

Communicated by E. J. Elston

T. Wang · R. L. Fernando (✉)
Department of Animal Sciences, University of Illinois, 1207 W. Gregory Drive, Urbana, IL 61801, USA

C. Stricker
Institute of Animal Sciences, Swiss Federal Institute of Technology, ETH-Zentrum CLU, CH-8092 Zürich, Switzerland

R. C. Elston
Department of Epidemiology and Biostatistics Western Reserve University, MetroHealth Medical Center, 2500 MetroHealth Drive, Cleveland, OH 44109, USA

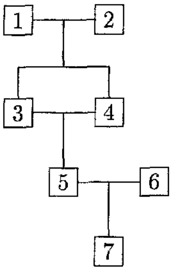


Fig. 1 A pedigree consisting of three families

ber of the pedigree, but it is possible for some members to have missing phenotypes. Thus a pedigree contains information on relationships between members and on a trait of interest. A memeber of the pedigree is a *founder* if neither of its parents are included in the pedigree (individuals 1 and 2), or a *terminal* member if it has no offspring included in the pedigree (individual 7). Note that a member, such as an isolated individual, can be both a founder and a terminal member.

A nuclear family, or simply family, is a pedigree consisting solely of both parents and their offspring. There are three nuclear families in Fig. 1: $F_1(1, 2; 3, 4)$ $F_2(3, 4; 5)$, and $F_3(5, 6; 7)$. A *connector* is a family member who belongs to at least one other family. For example, individuals 3 and 4 in $F_1(1, 2; 3, 4)$ are connectors. All family members in $F_2(3, 4; 5)$ are connectors. In $F_3(5, 6; 7)$, individual 5 is the only connector. A family is said to be *terminal* if it contains ony one connector. Family $F_3(5, 6; 7)$, for instance, is a terminal family. Two families are said to be *neighbors* to each other if they share at least one connector. Families $F_1(1, 2; 3, 4)$ and $F_2(3, 4; 5)$ are neighbors to each other through the two connnectors 3 and 4, and families $F_2(3, 4; 5)$ and $F_3(5, 6; 7)$ are neighbors to each other through the one connector 5.

The family in which an individual is an offspring is defined as the individual's anterior family, and the family in which an individual is a parent is defined as the individual's posterior family through the other parent. Individuals 1 and 2 in Fig. 1, for instance, have the posterior family $F_1(1, 2; 3, 4)$ through each other, and their anterior families are not included in the pedigree. Individual 3 has the anterior family $F_1(1, 2; 3, 4)$ and the posterior family $F_2(3, 4; 5)$ through mate 4. Individual 4 has the anterior family $F_1(1, 2; 3, 4)$ and the posterior family $F_2(3, 4; 5)$ through mate 3. Individual 5 has the anterior family $F_2(3, 4; 5)$ and the posterior family $F_3(5, 6; 7)$ through mate 6. Individual 6 has its poterior family $F_3(5, 6; 7)$ through mate 5.

An individual has only one anterior family (which is not part of the pedigree in the case of founders) and has a posterior family through each mate. A terminal member (e.g., individual 7) has no posterior family in the pedigree.

Computation of the likelihood for a pedigree without loops

For a pedigree without loops, Janss et al. (1992) proposed an iterative algorithm to compute the likelihood, whereas Fernando et al. (1993) proposed a non-iterative algorithm. We will present here a modified version of the iterative algorithm for a pedigree without loops, using the concepts of anterior and posterior probabilities given by Fernando et al. (1993). These will be referred to as the anterior and posterior values, because they do not sum to unity.

Iterative method

Following Fernando et al. (1993), the log likelihood for a pedigree without loops can be computed through any member i as

$$\log L(\mathbf{y}) = \log\left[\sum_{u_i} a_i(u_i) g(y_i|u_i) \prod_{j\varepsilon S_i} p_{ij}(u_i)\right] \tag{1}$$

where $\mathbf{y}$ is a vector of phenotypes (y_i) and, for individual i: u_i is the genotype, $a_i(u_i)$ is the anterior value, $g(y_i/u_i)$ is the penetrance value, $p_{ij}(u_i)$ is the posterior value through mate j, and S_i is the set mates. These quantities are functions of unknown parameters, and we denote by $L(\mathbf{y})$ the likelihood of these parameters for the data $\mathbf{y}$. To compute the anterior value for a member i, it is required to compute anterior and possibly posterior values for each of its parents and posterior values for each of its full sibs. To compute the posterior value for individual i through its mate j, it is required to compute anterior and posterior values for mate j and posterior values for each of their offspring (Fernando et al. 1993). This leads to an iterative and a recursive algorithm to compute the likelihood. In the computations shown below, an iterative algorithm is introduced where the anterior, posterior, and penetrance values are each scaled to sum to unity over genotypes in order to avoid numerical problems.

To obtain the unscaled value of the log likelihood, we accumulate the logs of the scaling factors used to compute anterior and posterior values for each member of the pedigree. For member i with mother m and father f, let K_{g_i} denote the log of the scaling factor for the penetrance value, $g(y_i|u_i)$; let K_{a_i} denote the accumulative log of the scaling factors used to compute the anterior value, $a_i(u_i)$; and let $K_{p_{ij}}$ denote the accumulative log for the posterior value through mate j, p_{ij}. These three log scaling factors are computed as

$$K_{g_i} = \log\left[\sum_{u_i} g(y_i|u_i)\right] \tag{2}$$

$$\begin{aligned} K_{a_i} = {} & K_{a_m} + K_{g_m} + \sum_{j\varepsilon S_m, j\neq f} K_{p_{mj}} + K_{a_f} + K_{g_f} \\ & + \sum_{j\varepsilon S_f, j\neq m} K_{p_{fj}} + \sum_{j\varepsilon C_{mf}, j\neq i} \left(K_{g_j} + \sum_{k\varepsilon S_j} K_{p_{jk}}\right) \\ & + \log\left[\sum_{u_i} a_i(u_i)\right] \end{aligned} \tag{3}$$

$$K_{p_{ij}} = K_{a_j} + K_{g_j} + \sum_{k\varepsilon S_j, k\neq i} K_{p_{jk}} + \sum_{k\varepsilon C_{ij}} \left(K_{g_k} + \sum_{l\varepsilon S_k} K_{p_{kl}}\right) + \log\left[\sum_{u_i} p_{ij}(u_i)\right] \qquad (4)$$

where $C_{ij}(C_{mf})$ is the set of offspring of parents i and j (m and f). Now the iterative algorithm to compute the log likelihood for a pedigree without loops is:

(1) For each member i:
(i) initialize anterior values: set anterior values equal to the population genotype frequencies and set the anterior log scaling factor equal to zero,
(ii) initialize posterior values: set posterior values equal to unity and set the posterior log scaling factor equal to zero, and
(iii) computer penetrance values and the corresponding log scaling factor.

(2) For each connector i:
(i) for families in which i is an offspring, compute its anterior value $a_i(u_i)$ from Fernando et al. (1993) and compute the log scaling factor K_{a_i} from (3), using current values of required quantities,
(ii) for families in which i is a parent, compute its posterior value through each mate j, p_{ij}, from Fernando et al. (1993) and compute the corresponding log scaling factor $K_{p_{ij}}$ from formula (4), using current values of the required quantities.

(3) Repeat step 2 (usually less than ten times) until each of the anterior and posterior values for each member has converged.

Now the log likelihood for a pedigree without loops can be computed through any connector i as

$$\log L(\mathbf{y}) = \log\left[\sum_{u_i} \hat{a}_i(u_i)\hat{g}(y_i|u_i)\prod_{j\varepsilon S_i}\hat{p}_{ij}(u_i)\right] + K_{a_i} + K_{g_i} + \sum_{j\varepsilon S_i} K_{p_{ij}} \qquad (5)$$

where $\hat{a}_i(u_i)$, $\hat{g}(y_i|u_i)$, and $\hat{p}_{ij}(u_i)$ are the scaled values of $a_i(u_i)$, $g(y_i|u_i)$, and $p_{ij}(u_i)$.

This method of computing the likelihood, where anterior and posterior values are iteratively updated as described above, will be called "iterative peeling".

Non-iterative method

Convergence of the iterative process described above does not depend on the sequence of computations. It is possible, however, to find a sequence to compute the anterior and posterior values such that convergence is achieved in one iteration. Such a sequence will be referred to as an optimum peeling sequence (OPS). We present below a method to determine such an OPS for a pedigree without loops. This method will be used to approximate the likelihood for a pedigree with loops.

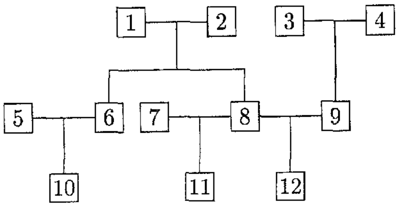
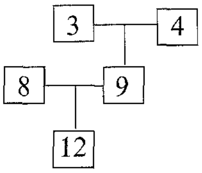


Fig. 2 A pedigree without loops consisting of five families

The first step to determine an OPS is to identify a terminal family. Recall that terminal family has only one connector. For an identified terminal family, compute its anterior value if the connector is an offspring or compute its posterior value through the other parent if the connector is a parent. The second step is to update the pedigree by removing all non-connector members of the terminal family from the pedigree. These two steps are repeated until only one family remains in the pedigree. Finally, choose an arbitrary member i from the last family; compute its anterior value if i is an offspring in the family or compute the posterior value for i through the other parent if i is a parent in the family. Note that if computations are in the sequence described above, quantities required to compute the anterior or posterior values will have been computed previously. Now the log likelihood is computed through i using (5). In the recursive computation of the likelihood (Fernando et al. 1993), an OPS is followed. The method described here is non-recursive.

To illustrate, consider the pedigree in Fig. 2. We begin the process of determining an OPS by arbitrarily choosing a terminal family, say F(5, 6; 10), in this pedigree. Because member 6 is the connector and also a parent in this family, we compute its posterior value through 5, $p_{6,5}$ (u_6), and update the pedigree by removing non-connector members 5 and 10 from the pedigree. In the updated pedigree, F(1, 2; 6, 8) becomes a terminal family, and member 8 is the connector and one of offspring. Thus, we compute the anterior value for member 8, $a_8(u_8)$, and update the pedigree by removing non-connector members 1, 2, and 6 from the pedigree. Now in the updated pedigree, F(7, 8; 11) becomes a terminal family, and member 8 is the connector and also a parent. Thus, we compute the posterior value for 8 through 7, $p_{8,7}$ (u_8), and remove non-connector members 7 and 11 from the pedigree. At this stage, there are only two families in the updated pedigree as shown below



In this remaining pedigree, F(8, 9; 12) becomes a terminal family, and member 9 is the connector and a parent in the family. Thus we compute the posterior value for 9 through 8, $p_{9,8}(u_9)$, and remove non-connector members 8 and 12 from the pedigree. The very last family in the updated pedigree is F(3, 4; 9), which has no connector. We choose an arbitrary offspring from the last family, in this case member 9, and compute its anterior value, $a_9(u_9)$. The likelihood is now computed through member 9 by equation (5), using the anterior and posterior values for member 9. Note that if computations are in the sequence described above, quantities required to compute the anterior or posterior values will have been computed previously. For example, to compute $a_8(u_8)$, $p_{6,5}$ (u_6) is required, but it has already been computed.

Completion of this process yields an OPS: [$p_{6,5}(u_6)$, $a_8(u_8)$, $p_{8,7}(u_8)$, $p_{9,8}(u_9)$, $a_9(u_9)$]. This sequence can be used repeatedly for likelihood computations such as in maximum-likelihood analysis. Note that an OPS is not unique. Sequence [$p_{8,7}(u_8)$, $p_{6,5}(u_6)$, $a_9(u_9)$, $p_{8,9}(u_8)$, $a_8(u_8)$], for example, is also an OPS.

In the above development, the pedigree was reduced in size by removing the non-connector members of a terminal family, one at a time, after computing the anterior or posterior values for the connector. This method of computing the likelihood will be known as "terminal peeling" to distinguish this peeling strategy from iterative peeling discussed in the previous section.

Terminal peeling is more efficient than iterative peeling because anterior or posterior values for each connector are computed only once in terminal peeling, whereas they are computed repeatedly in iterative peeling.

Terminal peeling can also be used to compute genotype probabilities for all members of the pedigree. First, the pedigree is reduced to a single family by terminal peeling. Second, the reduced pedigree is extended by adding families, which were removed in reverse order. As each family is added, the anterior and posterior values that have not already been computed are also computed. Finally, posterior genotype probabilities are computed, using the formula given by Fernando (1993), which leads to a non-recursive algorithm to compute the posterior genotype probability for each member in a pedigree without loops.

Approximation to the likelihood for a pedigree with loops

If a pedigree cannot be reduced to a single family by terminal peeling, then the remaining part of the pedigree has one or more loops. This remaining part with loops will be referred to as the "looped" part of the pedigree. There are some methods available to compute the exact likelihood for a pedigree with loops (Lange and Elston 1975; Cannings et al. 1978), but these methods are not suitable for livestock pedigrees with complex loops and thousands of individuals. The likelihood for a pedigree with loops can be approximated by introducing artificial individuals into the pedigree to cut each loop (a procedure to identify a loop in a pedigree is given in Appendix A). Introducing artificial individuals to cut loops results in a new "cut and extended" (cut-extended) pedigree, and an efficient algorithm for pedigrees without loops is then used to compute the likelihood for this cut-extended pedigree (Amos et al. 1986; Stricker et al. 1995). In the present paper, we improve this approximation to the likelihood for pedigrees with loops by three strategies: (1) computing the conditional likelihood for the cut-extended pedigree given the extended part, (2) cutting a loop by artificially introducing more than one individual into the pedigree, (3) choosing an optimum place to cut a loop.

These strategies are evaluated by comparing the improved approximation to the exact likelihood computed using MENDEL (Lange 1991).

Improvement by conditioning

Consider the looped pedigree consisting of two families shown in Fig. 3. This pedigree has an inbreeding loop due to the mating of 3 and 4. This loop can be arbitrarily cut by introducing an artificial founder 4*, with the same phenotype as 4, as the mate of 3 and the parent of 5. This results in the cut-extended pedigree shown in Fig. 4. An efficient algorithm that is appropriate for pedigrees without loops, such as terminal peeling, can now be applied to compute the likelihood for this cut-extended pedigree. This likelihood could be an approximation to the likelihood for the original pedigree with a loop. This approximation can be improved by conditioning the likehood on the phenotype of 4* , i.e., by computing the likelihood for the cut-extended pedigree in Fig. 4 and dividing it by the likelihood for individual 4* . Such an approximation is the conditional likelihood for the phenotypes of members 1 through 5 in the cut-extended pedigree, given the phenotype of individual 4*. The log likelihood for the entire cut-extended pedigree is -14.1912, whereas the log conditional likelihood is

Fig. 3 A pedigree with loops consisting of two families. The log likelihood was computed for a monogenic trait controlled by a single locus with two alleles, allele frequencies 0.6 and 0.4, genotypic means 10, 15, and 20, and a residual $\sim$ N(0, 10). The phenotypic values were: for individual 1, 20.0; for individual 2, 14.0; for individual 4, 17.0; and for individual 5, 22.0. The phenotype for individual 3 was missing

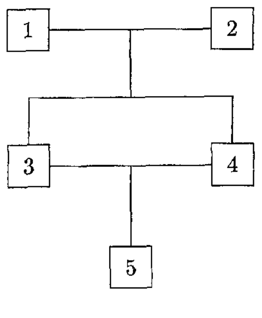


Fig. 4 A cut-extended pedigree for the looped pedigree in Fig. 3 after the first step of the iteration for connector 3

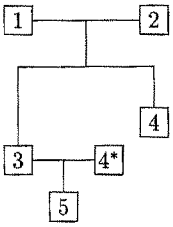


Fig. 5 A cut-extended pedigree for the looped pedigree in Fig. 3. The pedigree has been cut at individual 4 and extended by an artificial family

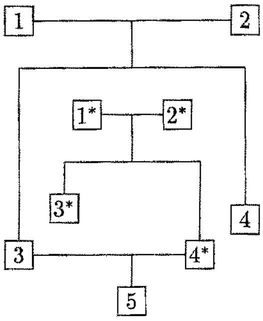
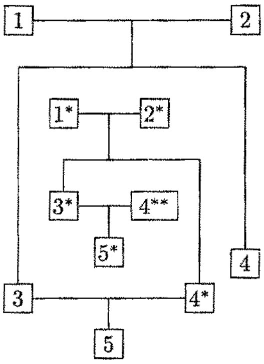


Fig. 6 A cut-extended pedigree for the looped pedigree in Fig. 3. The pedigree has been cut at individual 4 and extended by two successive artificial families



− 11.4787; the exact log likelihood for the pedigree in Fig. 3 is − 11.3515.

In general, for a pedigree with multiple and nested loops the procedure to approximate the likelihood is to:

(1) Identify each loop and cut it by introducing an artificial individual to obtain a cut-extended pedigree.
(2) Compute the likelihood for the entire cut-extended pedigree using terminal peeling.
(3) Compute the likelihood for the extended parts (i.e., for the artificially introduced individuals) by applying terminal peeling to the cut-extended pedigree after setting the phenotypes of original individuals in the cut-extended pedigree to be missing.
(4) Approximate the likelihood for a pedigree with loops by computing the likelihood for the cut-extended pedigree and dividing it by the likelihood for the extended parts.

Improvement by extending

In the cut-extended pedigree in Fig. 4, artificial individual 4*, a parent of 5, is a founder, whereas in the original pedigree (Fig. 3) individual 4 is a non-founder with parents 1 and 2 and with sib 3. Thus, the cut-extended pedigree in Fig. 4 is further extended by introducing an artificial family, resulting in the cut-extended pedigree in Fig. 5. In this cut-extended pedigree, each artificially introduced individual *i** is assigned the same phenotype as individual *i*. The likelihood is now approximated by computing the likelihood for the entire cut-extended pedigree and dividing it by the likelihood for the artificially introduced family. After extension by an artificial family, the approximate log likelihood for the pedigree in Fig. 3 is − 11.3818, which is closer to the exact log likelihood of − 11.3515 than that of − 11.4787 found previously when only one artificial individual was introduced.

For a pedigree with multiple loops, in general, each loop is cut arbitrarily by introducing an artificial family – the individual, its parents and full-sibs. The phenotype of each individual in an artificial family is the same as that of the corresponding original individual. The likelihood is then approximated by computing the likelihood for the entire cut-extended pedigree and dividing it by the likelihood for the artificially introduced families. The likelihood for the artificially introduced families is obtained by computing the likelihood for the cut-extended pedigree, after setting phenotypes of the original individuals to be missing. Likelihoods for the cut-extended pedigree and for the artificially introduced families are computed using the terminal peeling method.

The cut pedigree in Fig. 5 can be extended even further by introducing an artificial mate 4**, with the same phenotype as 4, for 3*, and an offspring 5* from the mating of 3* and 4** (Fig. 6). The approximate log likelihood obtained using this cut-extended pedigree is − 11.3506.

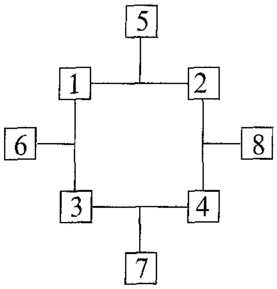
This process of extending the cut pedigree can be continued indefinitely. Let $\mathbf{y}_o$ be phenotypes of the original members in the cut-extended pedigree, and let $\mathbf{y}_{a_1}, \mathbf{y}_{a_2}, \ldots,$ be phenotypes of members of the successively introduced artificial families. Note that $\mathbf{y}_{a_i}$ is less related to $\mathbf{y}_o$ than $\mathbf{y}_{a_i - 1}$. Thus the conditional likelihood, $L(\mathbf{y}_o|\mathbf{y}_{a_1},\mathbf{y}_{a_2},\ldots)$, converges after several extensions. The log conditional likelihood for the looped pedigree in Fig. 3, for example, converged to − 11.3506 after the extension given in Fig. 6 (i.e., further extension did not change the fourth decimal digit of the log likelihood).

Improvement by optimum loop-cutting

A looped pedigree can be cut and extended at any connector. The accuracy of approximation to the likelihood obtained from the cut-extended pedigree, however, will depend for each loop on where the looped pedigree is cut and extended. Here we describe rules to cut and extend pedigrees that give a good approximation to the likelihood under a wide range of conditions. These rules were determined empirically and will not always yield the best approximation.

Recall that a family in a loop contains more than one connector, and cutting always occurs at connectors. There are two types of connectors in a loop. Type-I connectors are those with anterior and posterior families in the looped part of the pedigree, and type-II connectors are those with only posterior families in the looped part of the pedigree.

It is possible to have a loop with only type-II connectors. In Fig. 3, for example, members 3 and 4 are type-I connectors, whereas in the following pedigree



founders 1,2,3, and 4 are type-II connectors; there are no type-I connectors.

A loop with a type-I connector, say individual i, can be cut by replacing the connnector in its posterior family or families with an artificial individual i^*, while leaving the original connector in its anterior family as a terminal member. Starting at i^*, more artificial individuals can be introduced appropriately as described in the previous subsection. These artificially introduced individuals collectively are called the "anterior extension", and cutting a loop as described above will be referred to as "cutting with anterior extension".

A loop with a type-II connector can be cut by replacing the connector in one of its posterior families that is part of the loop with an artificial individual i^* as a founder. Starting at i^*, additional artificial individuals can be introduced, as described in the previous subsection. These artificially introduced individuals collectively are called the "posterior extension", and cutting a loop as described above will be referred to as "cutting with posterior extension".

The first step in the optimum loop-cutting procedure is to compute anterior and posterior values for all connectors by iterative peeling. Second, for each type-I connector i in a loop, compute

$$maxant = \frac{\max_{u_i}[\Pr(u_i|\mathbf{y}_a)]}{\max_{u_i}[\Pr(u_i|y_i)]} \qquad (6)$$

and for each connector i (both type-I and II) in a loop, through each mate j, compute

$$maxpost_j = \frac{\max_{u_i}[\Pr(u_i|\mathbf{y}_{p_j})]}{\max_{u_i}[\Pr(u_i|y_i)]} \qquad (7)$$

where $\Pr(u_i|\mathbf{y}_a)$, $\Pr(u_i|y_i)$, and $\Pr(u_i|\mathbf{y}_{p_j})$ are computed from population genotype frequencies $\Pr(u_i)$ and from the anterior, posterior and penetrance values, as shown below:

$$\Pr(u_i|\mathbf{y}_a) = \frac{a_i(u_i)}{\sum_{ui} a_i(u_i)}$$

$$\Pr(u_i|y_i) = \frac{\Pr(y_i|u_i)\Pr(u_i)}{\sum_{ui}\Pr(y_i|u_i)\Pr(u_i)}$$

$$\Pr(u_i|\mathbf{y}_{pj}) = \frac{\prod_j p_{ij}\Pr(u_i)}{\sum_{ui}\left[\prod_j p_{ij}\Pr(u_i)\right]}.$$

For individual i, the $\max_{u_i}[\Pr(u_i|\mathbf{y}_a)]$ measures the effect of the anterior extension on the genotype determination, $\max_{u_i}[\Pr(u_i|\mathbf{y}_P)]$ measures the effect of the posterior extension on the genotype determination, and $\max_{u_i}[\Pr(u_i|y_i)]$ measures the effect of the phenotype on the genotype determination. Relative to genotype determination for individual i by its own phenotype, the $maxpost_j$ measures the effects of the posterior extensions and the *maxant* measures the effects of the anterior extensions. Now for each loop:

(1) Find a type-I connector with the smallest *maxant* value. If any full-sib of this connector has been used to cut a lop, then cut the loop at this connector without anterior extension; otherwise cut the loop at this connector with anterior extension.
(2) If (1) fails, then among those type-II connectors that have not been used to cut a loop, find that with the smallest $maxpost_j$ value. Cut the loop at this connector with posterior extension.
(3) If both (1) and (2) fail, then cut the loop at an arbitrary type-II connector with posterior extension.

Algorithm to approximate the likelihood for a pedigree with loops

An efficient algorithm to approximate the likelihood for a pedigree with loops combines the above three improvements and terminal peeling. It uses the result from Appendix B that iterative peeling can be used to extend a pedigree with loops.

Briefly, first peel off terminal families successively until loops are found. Next apply iterative peeling to the looped part. Then use the optimum loop-cutting strategy to obtain the cut-extended pedigree. Now use terminal peeling to obtain the likelihood for the cut-extended pedigree. Compute the likelihood for the extended part. Finally the approximate likelihood for the original pedigree with loops is given by the likelihood for the cut-extended pedigree divided by the likelihood for the extended part.

In detail, the algorithm is:

(1) For each member i:
  (i) initialize its anterior and posterior values, and
  (ii) compute penetrance values and the corresponding log scaling factor.

(2) Apply terminal peeling to peel off terminal families successively until loops are encountered. Note that phenotypic information from terminal families is now accumulated into either anterior or posterior values for the connectors of families that are in the looped part of the pedigree.

(3) Apply iterative peeling on the looped part of the pedigree.

(4) Identify a loop and cut it using the optimum loop-cutting strategy. Determine an optimum peeling sequence (OPS) to peel off terminal families that result from loop-cutting. Repeat this process until all loops have been cut.

(5) For each member i in the cut-extended pedigree:
  (i) if i was chosen to be cut with anterior extension, then set the anterior values of i^* to be the iteratively computed anterior values of i,
  (ii) if i was chosen to be cut with posterior extension through mate j, then set the posterior values of i^* to be the iteratively computed posterior values of i through j,
  (iii) if i was chosen to be cut multiple times, say one cut with anterior extension and one with posterior extension through mate j, then set the anterior values of i * to be the iteratively computed anterior values of i and the posterior values of i^* to be the iteratively computed posterior values of i through j.

Now a cut-extended pedigree has been generated. Note that the extended parts of the cut-extended pedigree were generated in step (3) by iterative peeling (Appendix B).

(6) Use terminal peeling to compute the likelihood for the cut pedigree. Note that the iteratively computed anterior and posterior values for the artificially introduced individuals contribute to the likelihood. This gives the likelihood for the cut-extended pedigree.

(7) Set phenotypes of all original individuals to be missing, and apply terminal peeling again to compute the likelihood for the extended part.

(8) Finally, compute L $(\mathbf{y}_o|\mathbf{y}_{ext})$ as

$$L(\mathbf{y}_o|\mathbf{y}_{ext}) = \frac{L(\mathbf{y}_o, \mathbf{y}_{ext})}{L(\mathbf{y}_{ext})} \tag{8}$$

where $\mathbf{y}_o$ are phenotypes of original individuals in the cut-extended pedigree and $\mathbf{y}_{ext}$ are phenotypes of extended individuals. This conditional likelihood is the approximation to the likelihood for a pedigree with loops.

The algorithm described above will be referred to as the "eight-step" approximation.

Examples

Example 1

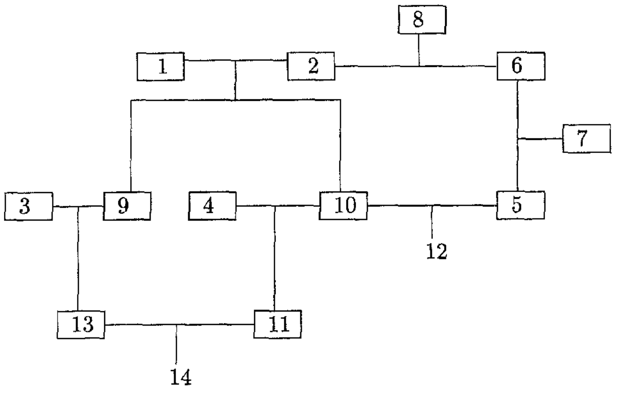
Consider a pedigree with loops consisting of seven families (Fig. 7). Phenotypes relate to a disease controlled by a single recessive allele with penetrance = 0.9. Individuals 12 and 14 are the only two members who are affected. The exact log likelihood for this pedigree was computed using MENDEL.

There are two loops in the pedigree. The upper-right loop is cut at member 10 in the family (1, 2; 9, 10) with anterior extension and the lower-left loop is cut at member 11 in the family (4, 10; 11) with anterior extension. The cut-extended pedigree, originating from the looped pedigree in Fig. 7, is in Fig. 8. This cut-extended pedigree will be referred to as *cut11*. The approximate log likelihoods of *cut11* were computed using multiple-individual anterior extension and single-individual extension (Stricker et al. 1995). These two approximations are referred to as *cut11-m* and *cut11-s* (Table 1).

We compare another cutting scheme where the upper-right loop is cut as in *cut11* and connector 13 in family (3, 9; 13) is chosen to cut the lower-left loop with anterior extension. The approximate log likelihood for this cut-extended pedigree with multipe-individual anterior extension was computed, and is referred to as *cut13*. The three approximations to the log likelihood, together with the exact value, are given in Table 1 for allele frequencies (p) ranging from 0.02 to 0.98. From Table 1 it can be seen that *cut11-m* is closer to the exact value than *cut13* (in fact *cut11-m* and the exact value are identical to four decimal places) and multiple-individual extension is always better than single-individual extension.

As stated earlier, the proposed optimum-loop cutting strategy does not always yield the best approximation. In this pedigree, for example, when $p < 0.35$, our rule chooses *cut11-m*, which is obviously better than *cut13*. When $p \geqslant 0.35$, however, our rule chooses *cut13*, though *cut11-m* is slightly better than *cut13* (Table 1).

Fig. 7 A pedigree with loops consisting of seven families for disease trait. Each affected individual is drawn without a framebox



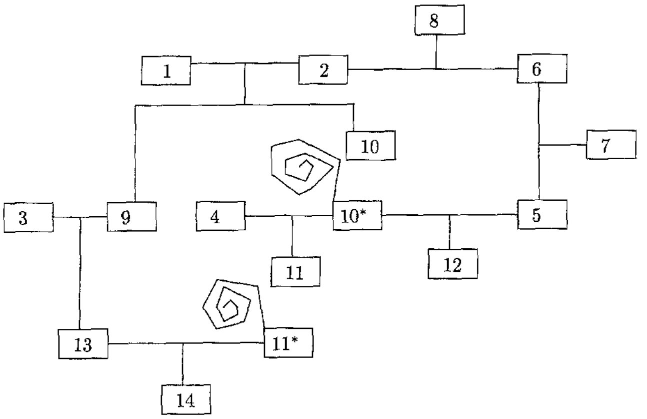


Fig. 8 A cut-extended pedigree for the looped pedigree in Fig. 7

To futher study this approach, we compare the "eight-step" approximations with the exact log likelihoods at two fixed allele frequencies, $p = 0.02$ and $p = 0.98$, for a range of penetrance values (Table 2). In order to obtain a better visualization of Tables 1 and 2 we plotted the log likelihoods versus allele frequencies (Fig. 9) and versus penetrance values (Fig. 10). It is clear from the two figures that the curves for the eight-step approximation and the exact log likelihood are vertically identical, though they need only be parallel for the approximation to be useful for maximum-likelihood estimation and hypothesis testing.

Example 2

To examine the accuracy of the eight-step approximation, an extremely looped pedigree consisting of 100 individuals with 63 loops is used. The structure of the pedigree can be described as follows: each of ten dams is mated to each of eight sires and there is one offspring from each of these 80 matings. We originally attempted to have ten sires as well as ten dams (100 matings) but MENDEL was not able to handle such a "large" pedigree on a SUN SparcStation2 with 46 megabytes of RAM. Thus two sires, as pedigree members, have no offspring.

The phenotypes were simulated for a monogenic trait with two alleles (*A*:*a*) with frequencies of 0.7:0.3, genotypic means of 15:10:5 for *AA*:*Aa*:*aa*, and a normally distributed residual with mean 0 and variance 2. The exact log likelihood using MENDEL is − 350.0287, whereas the approximation with the optimum loop-cutting and the iterative extension is − 349.9371, and with single individual extension it is − 345.2400. We also examined our proposed approximation for this looped pedigree with two more sets of simulated data resulting from two different residual variance values: 20 and 200. The exact log likelihoods using MENDEL for these two data sets were − 312.2798 and − 412.0731, respectively, whereas the approximated log likelihoods using the proposed eight-step approximation were − 311.6254 and − 412.1231.

Example 3

In order to demonstrate the feasibility of the eight-step approximation on a large pedigee, a monogenic trait, described in Example 2, was simulated for a pedigree of 10000 individuals with 780 loops. These loops were completely arbitrary and very complex.

We approximated the log likelihood for this large pedigree with complex loops. All computations were done on a SUN SparcStation2 with 46 megabytes of RAM. The exact log likelihood for this pedigree could not be calculated by MENDEL on this workstation. The CPU-time for the process of optimum loop-cutting was around 7 min, with an additional 5 min to compute log $L(\mathbf{y}_o | \mathbf{y}_{ext})$.

Discussion

We have presented an efficient algorithm to approximate the likelihood for large and complex pedigrees with loops by a combination of (1) conditioning the likelihood, (2) extending the pedigree, and (3) optimum loop-cutting with terminal peeling. For pedigrees without loops, the algorithm results in the exact likelihood. For pedigrees with loops, the algorithm resulted in good

Table 1 Comparison of approximations to the log likelihood with the exact value at different allele frequencies (penetrance = 0.9)

p	*exact*	*cut13*	*cut11-m*	*cut11-s*
0.02	− 11.5168	−9.5807	− 11.5168	− 15.9811
0.05	− 9.7233	− 8.5617	− 9.7233	− 12.5037
0.15	− 7.7659	− 7.3169	− 7.7659	− 8.7757
0.25	− 7.1331	− 6.8987	− 7.1331	− 7.4736
0.35	− 6.9947	− 6.8582	− 6.9947	− 6.9641
0.45	− 7.1891	− 7.1049	− 7.1891	− 6.9201
0.55	− 7.6856	− 7.6320	− 7.6856	− 7.2499
0.65	− 8.5237	− 8.4887	− 8.5237	− 7.9637
0.75	− 9.8311	− 9.8078	− 9.8311	− 9.1729
0.85	− 11.9490	− 11.9331	− 11.9490	− 11.2112
0.95	− 16.2184	− 16.2088	− 16.2184	− 15.4910
0.98	− 19.0845	− 19.0785	− 19.0845	− 18.5661

Table 2 Comparison of approximations to the log likelihood with the exact value at different penetrance values (allele frequencies $p = 0.02$ and $p = 0.98$)

Penetrance	$p = 0.02$		$p = 0.98$	
	exact	approx.	exact	approx.
0.1	− 15.7169	− 15.7169	− 5.9074	− 5.9026
0.2	− 14.3573	− 14.3573	− 5.8852	− 5.8779
0.3	− 13.5724	− 13.5724	− 6.6026	− 6.5910
0.4	− 13.0224	− 13.0224	− 7.7577	− 7.7390
0.5	− 12.6008	− 12.6008	− 9.2891	− 9.2598
0.6	− 12.2601	− 12.2601	− 11.1903	− 11.1508
0.7	− 11.9751	− 11.9751	− 13.4401	− 13.4045
0.8	− 11.7305	− 11.7305	− 15.9954	− 15.9796
0.9	− 11.5168	− 11.5168	− 19.0845	− 19.0845

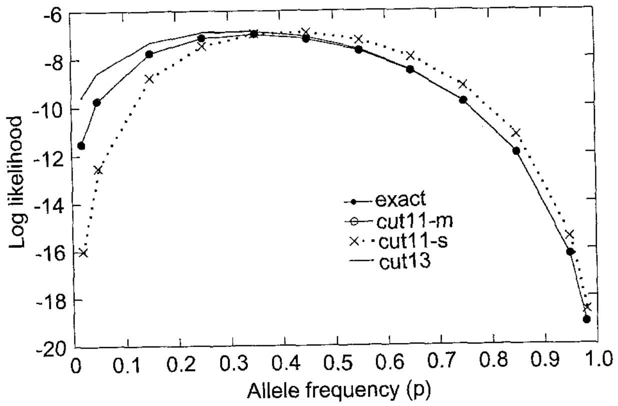


Fig. 9 Comparison of approximations to the log likelihood with the exact value at different allele frequencies (penetrance = 0.9)

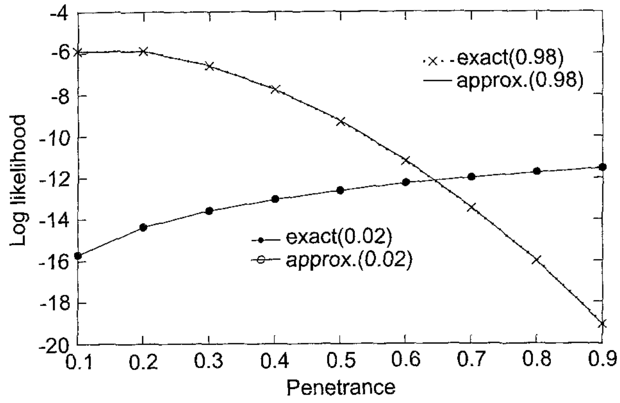


Fig. 10 Comparison of approximations to the log likelihood with the exact value at different penetrance values (allele frequencies $p = 0.02$ and $p = 0.98$)

approximations to the exact likelihoods for the cases examined in this paper. The proposed algorithm is feasible in terms of computing speed and memory requirements for livestock pedigrees with complex loops and thousands of individuals.

It is instructive to note that in Example 1 (Fig. 7) the *cut11-m* approximation results in virtually the exact log likelihood whenever the penetrance is high (0.9) or the disease allele frequency is low (0.02). These are just the situations where pedigree members 10 and 11 are almost certain to be heterozygotes, and cutting the pedigree at members for whom only one genotype is possible always leads to the exact log likelihood. When penetrance is low and the allele frequency is high, on the other hand, pedigree members 10 and 11 are more likely to be either heterozygotes or non-penetrant recessive homozygotes, and it is this ambiguity that causes the approximation.

Two options related to the qunatities *maxant* and *maxpost$_j$*, defined in (6) and (7), were also examined. The first was to substitute $\Pr(u_i|\mathbf{y})$ for $\Pr(u_i|y_i)$ in the denominators for *maxant* and *maxpost$_j$*. The second was to update *maxant* and *maxpost$_j$* before each loop-cutting when more than one cut was necessary. Based on limited testing using numerical examples, we have found that these two changes, in general, do not improve our proposed approximation to the likelihood for a pedigree with loops.

The terminal peeling presented in this paper can be used to compute efficiently the genotype probabilities for each member in a pedigree without loops (Fernando et al. 1993) and, using the approximation developed here, for each member in a pedigree with loops. The proposed algorithm can be applied to linkage analysis and marker-assisted selection, and the relative merits of the proposed eight-step approximation and Monte Carlo methods remain to be investigated.

Appendix

A. Procedure to identify a loop in the pedigree

To identify a loop in the pedigree: start by marking an arbitrary family, say F_i. Next mark one of F_i's neighboring families. Then continue to mark neighbors successively, avoiding to mark the same family twice. Because the pedigree has one or more loops, at some point it will not be possible to continue without marking a family twice. When a family, say F, is forced to be marked twice, a loop has been identified. The identified loop comprises all successively marked families between the first and the second marking of F, including F itself. For example, consider the pedigree in Fig. 7. Start by marking arbitrary family F(1, 2; 9, 10), which has three neighboring families. Next mark one of them, say F(3, 9; 13). The family F(3, 9; 13) has two neighboring families, F(1, 2; 9, 10) and F(13, 11; 14). Family F(1, 2; 9, 10) however has already been marked, so we mark F(13, 11; 14). This process continues, marking families F(4, 10; 11), F(10, 5; 12), F(5, 6; 7), and F(2, 6; 8). Family F(2, 6; 8) has two neighboring families F(1, 2; 9, 10) and F(5, 6; 7), each of which has been marked. At this point, it is impossible to mark a neighbor without marking a family twice. Thus we are forced to mark F(1, 2; 9, 10) again, and a loop has been identified. The loop comprises all successively marked families: F(1, 2; 9, 10), F(3, 9; 13), F(13, 11; 14), F(4, 10; 11), F(10, 5; 12), F(5, 6; 7), and F(2, 6; 8). After cutting this loop at a certain place, the process is repeated, as necessary, to identify further loops.

B. Extending a pedigree with loops by iterative peeling

Iterative peeling as described in this paper results in extending a pedigree with loops. Consider the pedigree with one loop in Fig. 3. First, we initialize anterior and posterior values for each member and compute the penetrance values and the corresponding log scaling factors. Then we compute the anterior and posterior values for connectors 3 and 4, sequentially.

In the first step of the iteration, computing the anterior values of connector 3, $a_3(u_3)$, requires the anterior values of its parents 1 and 2 and the posterior values of full-sib 4 through its mate 3, $p_{4,3}(u_4)$. The posterior values $p_{4,3}(u_4)$ have been initialized to ones as if full-sib 4 had no mate, and these values are used to compute $a_3(u_3)$. Then the posterior values of connector 3 through its mate 4, $p_{3,4}(u_3)$, are computed, requiring the anterior values of 4. The anterior values of 4 have been initialized as if connector 4 was a founder, and $p_{3,4}(u_3)$ is computed using these anterior values for connector 4. Thus the iteratively computed $a_3(u_3)$ and $p_{3,4}(u_3)$ in the first step of the iteration are identical to the $a_3(u_3)$ and $p_{3,4*}(u_3)$ obtained from the cut-extended pedigree in Fig. 4.

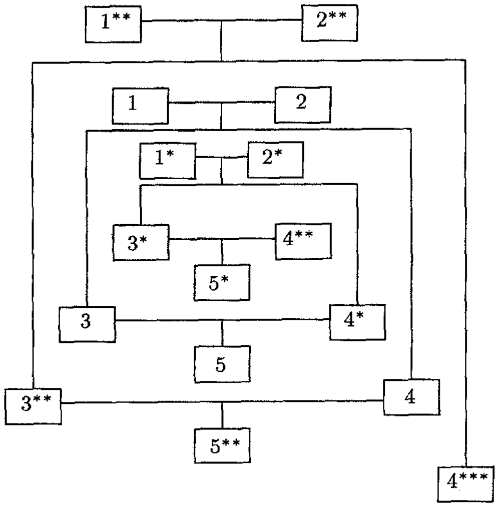
Still in the first step of the iteration, we need to compute the anterior values of connector 4. This computation requires the anterior values of its parents 1 and 2 and the posterior values of full-sib 3 through its mate 4. The posterior values $p_{3,4}(u_3)$ were computed during the first step of the iteration for connector 3, and these values are used to compute $a_4(u_4)$. Then the posterior values of connector 4 through its mate 3, $p_{4,3}(u_4)$, are computed. These values require the anterior values of individual 3, which were already computed. Thus the iteratively computed values $a_4(u_4)$ and $p_{4,3}(u_4)$ in the first step of the iteration are identical to the $a_4(u_4)$ and $p_{4*,3}(u_{4*})$ obtained from the cut-extended pedigree in Fig. 4.

In the second step of the iteration, computing $a_3(u_3)$ requires $a_1(u_1)$, $a_2(u_2)$, and $p_{4,3}(u_4)$. The required $p_{4,3}(u_4)$ was already computed in the first iteration using $a_3(u_3)$, which in turn used $a_1(u_1)$, $a_2(u_2)$, and the initialized $p_{4,3}(u_4)$. Then $p_{3,4}(u_3)$ is computed, requiring $a_4(u_4)$. The required $a_4(u_4)$ was already computed in the first iteration using $a_1(u_1)$, $a_2(u_2)$, and $p_{3,4}(u_3)$. Thus the anterior and posterior values $a_3(u_3)$ and $p_{3,4}(u_3)$ obtained in the second step of the iteration, are identical to the $a_3(u_3)$ and $p_{3,4*}(u_3)$ obtained from the cut-extended pedigree in Fig. 11.

Still in the second step of the iteration, we need to compute $a_4(u_4)$, which requires $a_1(u_1)$, $a_2(u_2)$, and $p_{3,4}(u_3)$. Again $p_{3,4}(u_3)$ was computed during the second step of the iteration for connector 3, and is now used to compute $a_4(u_4)$. Then $p_{4,3}(u_4)$ is computed, requiring $a_3(u_3)$. The required $a_3(u_3)$ in turn was computed during the second step of the iteration for connector 3. Thus the anterior and posterior values $a_4(u_4)$ and $p_{4,3}(u_4)$, obtained in the second step of the iteration, are identical to the $a_4(u_4)$ and $p_{4*,3}(u_{4*})$ obtained from the cut-extended pedigree in Fig. 11.

After further iterations, the cut-extended pedigree will look like a spiral (see Fig. 12): the more iterations, the more turns in the spiral. Thus, the results from iteratively peeling a looped pedigree are identical to those from the corresponding cut-extended pedigree.

Fig. 11 A cut-extended pedigree for the looped pedigree in Fig. 3 after the second step of the iteration for connectors 3 and 4



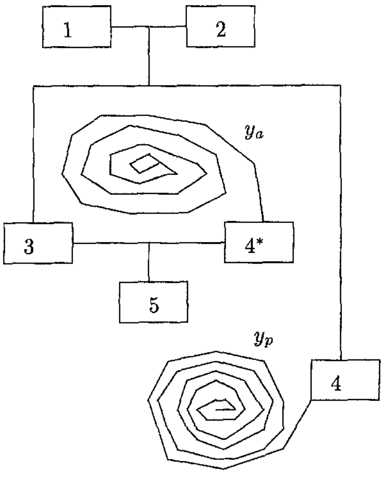
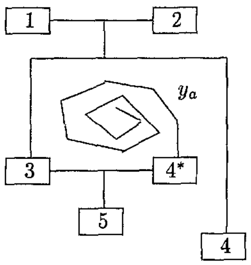


Fig. 12 A cut-extended pedigree for the looped pedigree in Fig. 3 after the second step of the iteration has been completed. Anterior and posterior extensions are drawn as coils

Fig. 13 A cut-extended pedigree for looped pedigree in Fig. 3 using an extension anterior to 4



Fortunately, no matter how much further a looped pedigree has been extended (see Fig. 12), we are able to split the cut-extended pedigree into three parts: $\mathbf{y}_o$, $\mathbf{y}_a$ and $\mathbf{y}_p$, where $\mathbf{y}_o$ are the phenotypes of the original members in the cut-extended pedigree (note that member 4 is no longer a mate of 3), $\mathbf{y}_a$ are the phenotypes of individuals introduced anterior to 4*, including the phenotype of 4*, and $\mathbf{y}_p$ are the phenotypes of individuals introduced posterior to 4*. This distinction enables the conditioning of the likelihood on the extended part of the looped pedigree. We have found that conditioning on one of these extensions is better than conditioning on all extensions available. Thus, after iterative peeling, loop cutting can be applied to the looped part of the pedigree. The iteratively computed anterior or posterior values are then attached to the artificially introduced individuals through which a loop is cut. Then, terminal peeling is appplied to peel off terminal families. From the above, it is obvious that cutting the looped part of a pedigree at different places results in different extended parts of the pedigree. Terminal peeling is now applied to the pedigree in Fig. 13.

Consider again the looped pedigree in Fig. 3, and assume that two steps of iterative peeling have been applied. Subsequently, the loop was cut at connector 4, which is thus duplicated, resulting in 4* with the same phenotype as 4. The anterior values of 4* are set to the iteratively computed anterior values for individual 4. Note that these iteratively computed anterior values correspond exactly to the anterior values for individual 4* in Fig. 11. Terminal peeling can now be applied to the pedigree in Fig. 13.

Acknowledgements This research was supported in part by the Illinois Agricultural Experiment Station, Hatch Project 35-0345 (R.L.F. and T.W.); the United States Department of Agriculture, grant No. 94-37205-1094 (R.L.F. and T.W.); U.S. Public Health

Service research grant GM 28356 from the National Institute of General Medical Sciences and resource grant 1P41PR03655 from the National Center for Research Resources (R.C.E.); and the Schweizerischer Nationalfonds, Switzerland (C.S.). The authors would like to thank an anonymous reviewer for instructive criticisms.

References

Amos CI, Wilson AF, Rosenbaum PA, Srinivasan SR, Webber LS, Elston RC, Berenson G (1986) An approach to the multivariate analysis of high-density lipoprotein cholesterol in a large kindred: The Bogalusa heart study. Genet Epidemiol 3:255–267

Cannings C, Thompson EA, Skolnick EH (1978) Probability functions on complex pedigrees. Adv Appl Prod 10:26–61

Elston RC, Stewart J (1971) A general model for the genetic analysis of pedigree data. Hum Hered 21:523–542

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

Guo SW, Thompson EA (1994) Monte Carlo estimation of mixed models for large complex pedigrees. Biometrics 50:417–432

Janss LLG, van der Werf JHG, van Arendonk JAM (1992) Detection of a major gene using segregation analysis in data from several generations. In: Proc Eur Assoc Anim Prod, Madrid, pp 144

Lange K (1991) Documentation for MENDEL, Version 3.0. Technical report, Department of Biomathematics, University of California, Los Angeles, California

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

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

Lange K, Matthysses S (1989) Simulation of pedigree genotypes by random walks. Am J Hum Genet 45:959–970

Lange K, Sobel E (1991) A random walk method for computing genetic location scores. Am J Hum Genet 49:1320–1334

Sheehan N, Thomas A (1993) On the irreducibility of a Markov chain defined on a space of genotype configuration by a sample scheme. Biometrics 49:163–175

Stricker C, Fernando RL, Elston RC (1995) An algorithm to approximate the likelihood for pedigree data with loops by cutting. Theor Appl Genet 91:1054–1063

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

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

Thomas DC, Cortessis V (1992) A Gibbs sampling approach to linkage analysis. Hum Hered 42:63–76

Thompson EA, Guo SW (1991) Evaluation of likelihood ratios for complex genetic models. IMA J Math Appl Med Biol 8:149–169

Thompson EA, Wijsman EML (1990) The Gibbs sampler on extended pedigrees: Monte Carlo methods for the genetic analysis of complex traits. Technical Report No. 193, Department of Statistics, University of Washington, Seattle, Washington