Theor Appl Genet (1995) 91:1054–1063

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

An algorithm to approximate the likelihood for pedigree data with loops by cutting

Received: 13 December 1994 / Accepted: 5 May 1995

Abstract This paper presents a recursive algorithm to approximate the likelihood in arbitrary pedigrees with loops. The algorithm handles any number and nesting levels of loops in pedigrees. The loops are cut as described in a previous publication and the approximate likelihood is simultaneously computed using the cut pedigree. No identification of a loop in the pedigree is necessary before the algorithm is applied.

Key words Arbitrary pedigrees with loops · Recursive algorithm · Peeling · Likelihood · Cutting loops

Introduction

Maximum likelihood is a powerful method for segregation and linkage analysis using pedigree data. We define a pedigree as a set of related individuals that can be represented graphically with lines connecting mates with each other and offspring with parents. Each member of a pedigree has either two parents in the pedigree or is a founder, i.e., has no parents in the pedigree. If an individual originally has only one known parent in the pedigree, this criterion can be satisfied by inserting an extra parent of unknown phenotype. A pedigree is said to have a loop when, in its graphical representation, it is possible to trace a continous path of connecting lines from one pedigree member to another, and then back to the same member by a different path. A more formal explanation is given in the Appendix. Such loops are common in large pedigrees, especially in livestock. If a pedigree has no loops, the likelihood for an oligogenic trait can be efficiently calculated following the principles of the Elston-Stewart algorithm (Elston and Stewart 1971; Fernando et al. 1993).

Lange and Elston (1975) showed how to calculate the likelihood for a pedigree with loops using the Elston-Stewart algorithm. Their strategy requires identifying each loop before calculating the likelihood. Each identified loop is 'cut' by introducing an additional founder into the pedigree such that a pedigree without loops is created. The exact likelihood is then calculated using a modified Elston-Stewart algorithm. Cannings et al. (1978) described a recursive algorithm to compute the exact likelihood for pedigrees with loops, implicitly requiring the identification of all loops in order to determine in what (peeling) sequence to perform the calculations. Thomas (1986) showed how it is possible to construct a good sequence automatically, comparing different methods of doing so. Another method to compute the exact likelihood that does not require identification of loops was given by Lange and Boehnke (1983). However, whichever method is used to calculate the exact likelihood, the number of computations or the amount of storage required is expected to be exponentially related to the number of nested loops in the pedigree. Thus, these methods may not be suitable for application to large complex pedigrees encountered in livestock. Recently, Monte Carlo methods have been used to estimate genetic parameters for complex human pedigrees (Sheehan 1989; Thompson and Wijsman 1990; Kong 1991; Thompson and Guo 1991). The application of Monte Carlo methods to livestock pedigrees needs to be investigated with respect to the feasibility and accuracy of the Monte Carlo approximation in the large pedigrees encountered in livestock. An iterative method to compute the likelihood for pedigrees without loops has been described by Janss et al. (1992). Their method

---

Communicated by E. J. Eisen

C. Stricker(✉)[1] · R. C. Elston
Department of Biometry and Genetics and the Center for Molecular and Human Genetics, Louisiana State University, Medical Center, 1901 Perdido Street, New Orleans 70112-1393, USA

R. L. Fernando
Department of Animal Sciences, University of Illinois at Urbana-Champaign, 1207 West Gregory Drive, Urbana IL 61801, USA

Present address:
[1] Institute of Animal Sciences, Swiss Federal Institute of Technology, ETH-Zentrum, CLU, CH-8092 Zürich, Switzerland

can be used to approximate the likelihood for large and complex pedigress with loops.

Another approach that has been used to approximate the likelihood for pedigrees with loops is to cut the pedigree as in Lange and Elston (1975) and use an algorithm applicable to pedigrees without loops. This strategy has been applied to human pedigrees with loops (e.g., Amos et al. 1986). At present, cutting a pedigree and ignoring the information about loops is accomplished by visually inspecting the pedigree and creating a cut pedigree without loops. Then the likelihood is calculated from the cut pedigree. This strategy cannot be applied to large and complex pedigrees because visual inspection of such pedigrees is not feasible. In general, the larger the number of individuals that form a loop, the less information is lost by cutting the loop and disregarding the relationship between the individuals that are cut apart. The objective of this paper is to present an algorithm to cut a pedigree with loops without the need for visual inspection, and to compute an approximate likelihood from the cut pedigree. The approach taken is to modify the algorithm presented by Fernando et al. (1993) to compute the likelihood for a pedigree without loops; from here on, this algorithm will simply be referred to as the FSE algorithm.

Recursive algorithm for pedigrees without loops

With respect to any member i of a pedigree without loops, the remaining members can be classified as anterior individuals or posterior individuals. The anterior individuals are connected to i through its parents m and f or its full sibs, the posterior individuals through its mates and corresponding offspring.

For a pedigree without loops, following Fernando et al. (1993), the likelihood is recursively computed as

$$L=\sum_{u_i} a_i(u_i)g(y_i|u_i)\prod_{j\in S_i} p_{i,j}(u_i), \tag{1}$$

where $a_i(u_i)$ is the anterior probability of individual i having genotype u_i, y_i is the phenotypic value of i, $g(y_i|u_i)$ is the penetrance function, $p_{i,j}(u_i)$ is the posterior probability for i through mate j for genotype u_i, and S_i is the set of mates of individual i.

The recursive process to calculate the anterior probability of individual i having genotype u_i, $a_i(u_i)$, is initiated by computing the following probabilities in the specified order:

(1) anterior probabilities $a_m(u_m)$ and $a_f(u_f)$ for m and f, the parents of i,
(2) posterior probabilites $p_{m,j_m}(u_m)$ and $p_{f,j_f}(u_f)$ for all mates j_m of m and j_f of f, except $j_m=f$ and $j_f=m$,
(3) posterior probabilities $p_{k,j_k}(u_k)$ for all offspring k of m and f, except $k=i$, through mates j_k of k.

The anterior probability $a_i(u_i)$ is eventually calculated as

$$\begin{aligned}a_i(u_i)=\sum_{u_m}\Bigg\{&a_m(u_m)g(y_m|u_m)\prod_{\substack{j\in s_m\\ j\neq f}} p_{m,j}(u_m)\\ &\times\sum_{u_f}\Bigg\{a_f(u_f)g(y_f|u_f)\prod_{\substack{j\in s_f\\ j\neq m}} p_{f,j}(u_f)\\ &\times tr(u_i|u_m,u_f)\\ &\times\prod_{\substack{k\in C_{mf}\\ k\neq i}}\Bigg\{\sum_{u_k} tr(u_k|u_m,u_f)g(y_k|u_k)\\ &\times\prod_{j\in s_k} p_{k,j}(u_k)\Bigg\}\Bigg\}\Bigg\},\end{aligned} \tag{2}$$

where $tr(u_i|u_m,u_f)$ is the transition probability for offspring i and parents m and f. This completes the recursive process to obtain $a_i(u_i)$. Note that if i is a founder, the anterior probabilities are given by the appropriate population genotype proportions.

The recursive process to calculate the posterior probability $p_{i,j}(u_i)$ of individual i through mate j, for genotype u_i, is initiated by computing the following probabilities in the specified order:

(1) anterior probabilities $a_j(u_j)$ for the mate j of i,
(2) posterior probabilities $p_{j,l_j}(u_j)$ for j through all the mates l_j of j, except $l_j=i$,
(3) posterior probabilities $p_{k,l_k}(u_k)$ for all offspring k of i and j, through the mates l_k of k.

Then, the recursive process to obtain $p_{i,j}(u_i)$ is completed by calculating

$$\begin{aligned}p_{i,j}(u_i)=\sum_{u_j}\Bigg\{&a_j(u_j)g(y_j|u_j)\prod_{\substack{l\in s_j\\ l\neq i}} p_{j,l}(u_j)\\ &\times\prod_{k\in C_{ij}}\Bigg\{\sum_{u_k} tr(u_k|u_i,u_j)g(y_k|u_k)\prod_{l\in S_k} p_{k,l}u_k\Bigg\}\Bigg\}\end{aligned} \tag{3}$$

Finally, the likelihood can be computed by equation (1).

Computing the likelihood for pedigrees with loops

If the FSE algorithm is applied to a pedigree with loops, recursive computation of the anterior or posterior probability for any individual in a loop will require the calculation of the same probability an infinite number of times, so that the recursive process is never completed. This is due to the fact that, with respect to any individual in a loop, the distinction between anterior and posterior individuals cannot be made. Two concepts need to be introduced in order to explain how the FSE algorithm can be modified to approximate the likelihood for a pedigree with loops. The first is the concept of anterior and posterior families of an individual. An individual i

can be an offspring of only one nuclear family. This is called the anterior family of i. On the other hand, i can be a parent in several nuclear families, and these are called the posterior families of i. The second concept is that of 'visiting' a nuclear family. A nuclear family is said to be visited if the recursive process to compute the anterior probability of an offspring of that nuclear family has been initiated, or if the recursive process to compute the posterior probability for one parent through the other parent of that nuclear family has been initiated.

In the FSE algorithm, the recursive process to compute an anterior or posterior probability can be prevented from seeking to compute the same probability as one that has previously been sought by ensuring that a nuclear family is not visited more than once. From equation (2), it can be seen that if the recursive process to compute the anterior probability for an individual is initiated, i.e., the anterior family for the individual has been visited, all information from its anterior family and all other nuclear families connected to this family will eventually contribute to the anterior probability. Similarly, it can be seen from equation (3) that if the recursive process to compute the posterior probability for an individual through a specific mate is initiated, all information from this specific posterior family and all other nuclear families connected to this family will eventually contribute to this posterior probability. It is explained below how to complete the anterior and posterior recursive processes, when a nuclear family that has already been visited is encountered, such that the likelihood for a cut pedigree is obtained.

Modifications to the calculation of an anterior probability

There can be at most three types of nuclear families, adjacent to the anterior family of individual i, that are visited in calculating the anterior probabilities for i. On initiating the recursive process in a pedigree with loops, any of these adjacent families may have already been visited, and so we need to ensure that none of them is revisited. The three types of families are:

(1) the anterior family of each parent m and f of i,
(2) the posterior families of m through each of the mates of m, except f, and the posterior families of f through each of the mates of f, except m,
(3) the posterior families of each of the sibs of i through their mates.

To avoid revisiting any of these families adjacent to the anterior family of individual i, we need to impose three rules, which will be numbered 1–3 corresponding to the type of the adjacent family involved.

Rule 1: before initiating the recursive process to compute the anterior probabilities of a parent, say m, we need to examine if the anterior family of m has already been visited. If it has, the recursive process to compute the anterior probabilities of m is not initiated. Instead, the recursive process to compute the anterior probabilities of individual i is modified by introducing an artificial founder m^* as a new parent for the anterior family of i. The founder m^* is assigned population genotype proportions as its anterior probabilities and is given the same phenotype as individual m. This artificial founder m^* has individual f as its single mate. All the posterior families of parent m other than through parent f are disregarded when parent m^* is newly created and introduced to the pedigree, because if an anterior family of a parent has been visited, all its posterior families will eventually be visited through the recursive process. Introducing m^* results in cutting a loop in the pedigree.

Rule 2: before initiating the recursive process to compute the posterior probabilities of a parent of i, say m, through a mate $j_m \neq f$, we need to examine if this posterior family of m through j_m has already been visited. If it has, the recursive process to compute the posterior probabilities for m through j_m is not initiated. Instead the recursive process to compute the anterior probabilities of individual i is modified by disregarding the posterior family of m through j_m.

Rule 3: before initiating the recursive process to compute the posterior probabilities of a sib k of individual i through a mate j_k, we need to examine if the posterior family of sib k through its mate j_k has already been visited. If this family has already been visited, the recursive process to compute the posterior probabilities for k through j_k is not initiated. Instead the recursive process to compute the anterior probabilities of individual i is modified by disregarding the posterior family of k through j_k.

Modifications to the calculation of a posterior probability

There can be at most three types of nuclear families, adjacent to the posterior family of individual i through a mate j, that are visited in calculating the posterior probabilities for i through j. Initiating the recursive process in a pedigree with loops, any of these adjacent families may have already been visited, and so we need to ensure that none of them is revisited. The three types of families are:

(1) the anterior family of the mate j,
(2) the posterior families of mate j through each of the mates of j, except i,
(3) the posterior families of k, the offspring of i and j, through each of the mates of k.

To avoid revisiting any of these families adjacent to the posterior family of individual i through mate j, we need to impose four rules which will be numbered 4–7. Rule 4 refers to the first type of adjacent family, rules 5 and 6 to the second type, and rule 7 to the last type in the above list.

Rule 4: before initiating the recursive process to compute the anterior probabilities of the mate j, we need to examine if the anterior family of j has already been visited. If it has been visited, the recursive processes to compute the anterior probabilities of j is not initiated. Instead, the recursive process to compute the posterior probabilities of individual i through mate j is modified by introducing an artificial founder j^* as a new parent for the posterior family of i through j. As for rule 1, the founder j^* is assigned population genotype proportions as its anterior probabilities and is given the same phenotype as individual j. This artificial founder j^* has individual i as its single mate, because if an anterior family of a mate has been visited, all its posterior families will eventually be visited through the recursive process. Introducing j^* results in cutting a loop in the pedigree.

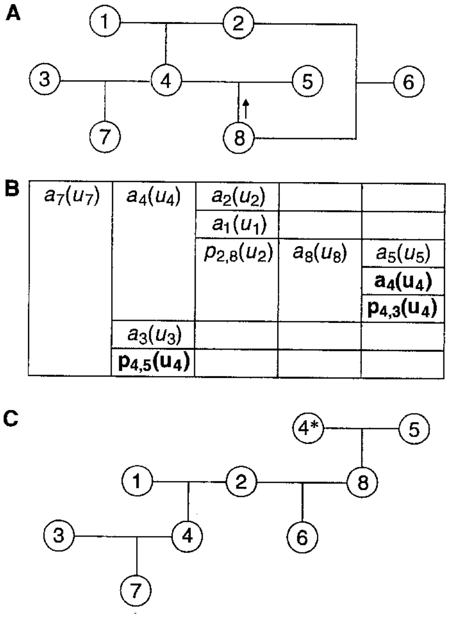
Rule 5: before initiating the recursive process to compute the posterior probabilities of mate j through one of js mates $l_j \neq i$, we need to examine if any of the posterior families of j through mates $l_j \neq i$ has already been visited. If any of these has been visited, none of the recursive processes to compute the posterior probabilities through the mates $l_j \neq i$ of j are initiated. Instead, the recursive process to compute the posterior probabilites of individual i through mate j is modified by introducing an artificial founder j^* as a new parent for the posterior family of i through j in a manner analogous to that described in rules 1 and 4. This artificial founder j^* has individual i as its single mate, because if any one of mate js posterior families has been visited, then all its posterior families will eventually be visited through the recursive process. Introducing j^* by rule 5 is equivalent to the cut performed by rule 4. Furthermore, rule 5 need only be applied to mates that have no parents, because the mate's anterior family, if it exists, will have already been verified by rule 4.

Rule 6: If no cut to the pedigree is introduced either by rule 4 or rule 5 before initiating the recursive process to compute the posterior probabilities of mate j through its mate $l_j \neq i$, we need to examine again if this specific posterior family of j through mate $l_j \neq i$ has already been visited. If it has been visited, the recursive process to compute the posterior probabilities of mate j through its mate $l_j \neq i$ is not initiated. Instead, the recursive process to compute the posterior probabilities of individual i through mate j is modified by disregarding the posterior family of mate j through its mate l_j. This rule is necessary because an individual may be part of several posterior families. By initiating the recursive process to compute the posterior probabilities for mate j through a specific mate $l_a \neq i$ of j, it is possible to visit another posterior family of j through mate $l_b \neq i$ of j.

Rule 7: before initiating the recursive process to compute the posterior probabilities of k, an offspring of the parents i and j, through its mate l_k, we need to examine if the posterior family of offspring k through its mate l_k has already been visited. If this family has already been visited, the recursive process to compute the posterior probabilities for k through l_k is not initiated. Instead, the recursive process to compute the posterior probabilities of individual i through mate j is modified by disregarding the posterior family of k through l_k.

Illustrative examples to rules 1–7

To illustrate rule 1, consider the calculation of the anterior probabilities of individual 7 in Fig. 1, part A. As shown in part B of Fig. 1, this requires calculation of the anterior probabilities of individuals 4 and 3, and the posterior probabilities of individual 4 through 5. Individual 3 is a founder, thus its anterior probabilities are obtained without further recursion. To compute the anterior probabilities of individual 4, the anterior probabilities of its parents 2 and 1 are required, as well as the posterior probabilities of individual 2 through its mate 8 [$p_{2,8}(u_2)$]. Individuals 1 and 2 are founders, thus no further recursion is required. To obtain $p_{2,8}(u_2)$, the anterior probability of individual 8 [$a_8(u_8)$] is required. To compute $a_8(u_8)$, we require $a_5(u_5)$, $a_4(u_4)$ and $p_{4,3}(u_4)$. Individual 5 is a founder, and no further recursion is needed. However, the recursive process to compute $a_4(u_4)$ is not initiated (by rule 1) as its anterior family has already been visited. Instead, the recursive process to compute $a_8(u_8)$ is modified by introducing an artificial founder 4* as a new parent for the anterior family of 8. The artificial founder 4* is given population genotype proportions as anterior probabilities, the same phenotype as individual 4, and has individual 5 as its single mate. Rule 1 also ensures that $p_{4,3}(u_4)$ is not initiated, because individual 4* is completely cut off from all



$a_7(u_7)$	$a_4(u_4)$	$a_2(u_2)$		
		$a_1(u_1)$		
		$p_{2,8}(u_2)$	$a_8(u_8)$	$a_5(u_5)$
				$a_4(u_4)$
				$p_{4,3}(u_4)$
	$a_3(u_3)$			
	$p_{4,5}(u_4)$			

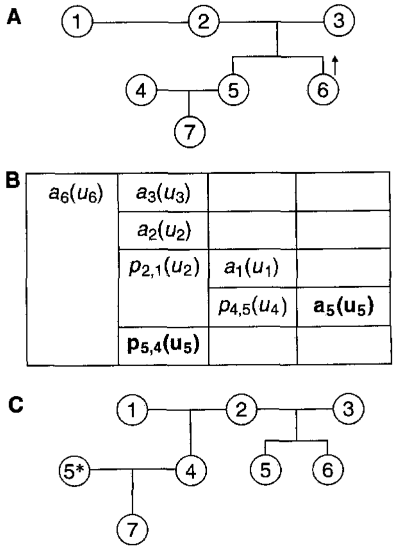
Fig. 1 A Pedigree used to illustrate modifications to the recursive process to compute anterior probabilities when the anterior or posterior family of a parent has been visited. The direction of the start of the recursive process is indicated by an *arrow*. **B** Probabilities required to recursively calculate $a_7(u_7)$ in the pedigree of part A. Each column gives the probabilities required to calculate those in the column to the left of it. The recursive process to compute the probabilities printed in *bold-face* are not initiated. **C** Cut pedigree resulting from the application of the modified algorithm to the pedigree in part A

posterior families other than through 5, which we are currently visiting as the anterior family of individual 8. Thus, the computation of $a_8(u_8)$ is completed with 5 and 4* as its parents. In this example, the recursive process to compute $a_8(u_8)$ is modified because the anterior family of one of its parents was already visited. Then, $a_8(u_8)$ is used to calculate $p_{2,8}(u_2)$ and this in turn is used to calculate $a_4(u_4)$. Computing $a_8(u_8)$ in this way is equivalent to cutting the loop in part A of Fig. 1 and computing $a_8(u_8)$ from the cut pedigree in part C of Fig. 1.

To illustrate rule 2, we continue the computation of $a_7(u_7)$ in Fig. 1. As mentioned before, computing $a_7(u_7)$ requires the posterior probabilities of individual 4 through mate 5 $[p_{4,5}(u_4)]$. However, the recursive process to compute $p_{4,5}(u_4)$ is not initiated (by rule 2), because this posterior family through mate 5 has already been visited to compute $a_8(u_8)$. Thus, the recursive process to compute $a_7(u_7)$ is modified by disregarding the posterior family of individual 4 through mate 5. Now, the recursive process to calculate $a_7(u_7)$ can be completed. Computing $a_7(u_7)$ this way is equivalent to cutting the loop in part A of Fig. 1 and calculating $a_7(u_7)$ from the cut pedigree given in part C of Fig. 1.

To illustrate rules 3 and 4, consider the calculation of the anterior probabilities of individual 6 in part A of Fig. 2. As shown in part B of Fig. 2, this requires calculation of the anterior probabilities of parents 3 and 2, the posterior probabilities of parent 2 through its mate 1 and the posterior probabilities for sib 5 through its mate 4. The recursive process to compute $p_{2,1}(u_2)$ requires $a_1(u_1)$ and $p_{4,5}(u_4)$; and the computation of $p_{4,5}(u_4)$ requires $a_5(u_5)$ (part B of Fig. 2). However, the recursive process to compute $a_5(u_5)$ is not initiated by rule 4, because the anterior family of individual 5 has already been visited when $a_6(u_6)$ was initially sought. Instead, the recursive process to compute $p_{4,5}(u_4)$ is modified by introducing an artificial founder 5* as a new parent for the posterior family of 4 through 5. The artificial founder 5* is given population genotype proportions as anterior probabilities, the same phenotype as individual 5, and has individual 4 as its single mate. Now the computation of $p_{4,5}(u_4)$ and subsequently $p_{2,1}(u_2)$ can be completed. Computing $p_{4,5}(u_4)$ in this manner is equivalent to cutting the loop in part A of Fig. 2 and calculating $p_{4,5^*}(u_4)$ from the cut pedigree given in part C of Fig. 2. In this example, the recursive process to compute $p_{4,5}(u_4)$ was modified (by rule 4), because the anterior family of individual 5 was already visited. As mentioned before, $p_{5,4}$ is also required to complete the computation of $a_6(u_6)$. However, the recursive process to compute $p_{5,4}(u_4)$ is not initiated by rule 3, because the posterior family of individual 5 through mate 4 has already been visited by computing $p_{4,5}(u_4)$ in order to obtain the previously required $p_{2,1}(u_1)$. Instead, the recursive process to compute $a_6(u_6)$ is modified by disregarding the posterior family of 5 through 4. Now the recursive process to compute $a_6(u_6)$ can be completed. Computing $a_6(u_6)$ in this way is equivalent to cutting the loop in Fig. 2 and calculating $a_6(u_6)$ from the cut pedigree given in part C of Fig. 2.

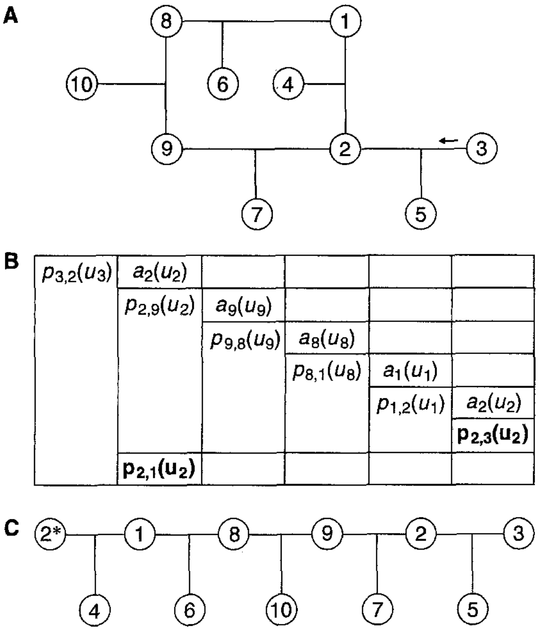
Fig. 2 **A** Pedigree used to illustrate modifications to the recursive process to compute anterior probabilities when the posterior family of a sib has been visited. The direction of the start of the recursive process is indicated by an *arrow*. **B** Probabilities required to recursively calculate $a_6(u_6)$ in the pedigree of part A. Each column gives the probabilities required to calculate those in the column to the left of it. The recursive process to compute the probabilities printed in *bold-face* are not initiated. **C** Cut pedigree resulting from the application of the modified algorithm to the pedigree in part A



$a_6(u_6)$	$a_3(u_3)$		
	$a_2(u_2)$		
	$p_{2,1}(u_2)$	$a_1(u_1)$	
		$p_{4,5}(u_4)$	$\mathbf{a_5(u_5)}$
	$\mathbf{p_{5,4}(u_5)}$		

To illustrate rules 5 and 6, consider computing the posterior probabilities of individual 3 through mate 2 in part A of Fig. 3. As shown in part B of Fig. 3, the recursive process to compute $p_{3,2}(u_3)$ eventually requires $p_{1,2}(u_1)$. Because mate 2 is a founder, population genotype proportions are assigned as anterior probabilities, i.e., no anterior family can be verified. Then $p_{2,3}(u_2)$ is required. However, the recursive process to compute $p_{2,3}(u_2)$ is not initiated by rule 5, because the posterior family of individual 2 through mate 3 has already been visited to obtain $p_{3,2}(u_3)$. Instead, the recursive process to compute $p_{1,2}(u_1)$ is modified by introducing an artificial founder 2* as a new parent for the posterior family of individual 1 through mate 2. The artificial founder 2* is given population genotype proportions as anterior probabilities and the same phenotype as individual 5 according to rule 5. Now, the computation of $p_{1,2}(u_1)$ and subsequently $p_{2,9}(u_2)$ can be completed. The final computation of $p_{3,2}(u_3)$ will also require $p_{2,1}(u_2)$, as shown in part B of Fig. 3. However the recursive process to compute $p_{2,1}(u_2)$ is not initiated by rule 6, because this posterior family of individual 2 has already been visited to obtain $p_{1,2}(u_1)$. Now $p_{3,2}(u_3)$ can eventually be computed. Computing $p_{3,2}(u_3)$ in this manner is equivalent to cutting the loop in part A of Fig. 3 and calculating $p_{3,2}(u_3)$ from the cut pedigree given in part C of Fig. 3.

To illustrate rule 7 (and rule 4), consider the calculation of the posterior probabilities of individual 1



$p_{3,2}(u_3)$	$a_2(u_2)$				
	$p_{2,9}(u_2)$	$a_9(u_9)$			
		$p_{9,8}(u_9)$	$a_8(u_8)$		
			$p_{8,1}(u_8)$	$a_1(u_1)$	
				$p_{1,2}(u_1)$	$a_2(u_2)$
					$\mathbf{p_{2,3}(u_2)}$
	$\mathbf{p_{2,1}(u_2)}$				

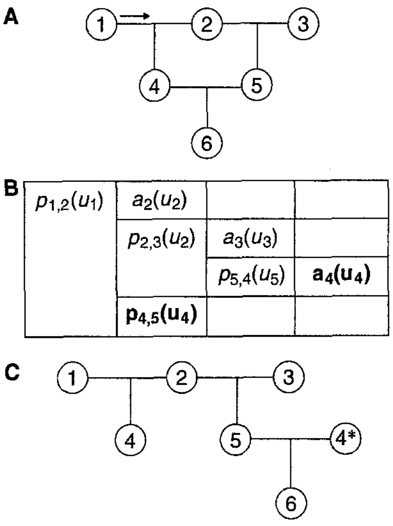
Fig. 3 A Pedigree used to illustrate modifications to the recursive process to compute posterior probabilities when the anterior or posterior family of a parent has been visited. The direction of the start of the recursive process is indicated by an *arrow*. **B** Probabilities required to recursively calculate $p_{3,2}(u_3)$ in the pedigree of part A. Each column gives the probabilities required to calculate those in the column to the left of it. The recursive process to compute the probabilities printed in *bold-face* are not initiated. **C** Cut pedigree resulting from the application of the modified algorithm to the pedigree in part A

through mate 2 in part A of Fig. 4. As shown in part B of Fig. 4, the pedigree is first cut by rule 4, a modification to $p_{5,4}(u_5)$, because the anterior family of individual 4 has already been visited when $p_{1,2}(u_1)$ was initially sought. Finally, $p_{4,5}(u_4)$ is required to complete the computation of $p_{1,2}(u_1)$. The recursive process to compute $p_{1,2}(u_1)$ is modified, according to rule 7, by disregarding the posterior family of individual 4 through mate 5, because it has already been visited. Now, the recursive process to compute $p_{1,2}(u_1)$ can be completed. Computing $p_{1,2}(u_1)$ as above is equivalent to cutting to loop in part A of Fig. 4 and calculating $p_{1,2}(u_1)$ from the cut pedigree given in part C of Fig. 4.

Calculating genotype probabilities in pedigrees with loops

The algorithm presented here to approximate the likelihood in arbitrary pedigrees with loops can also be used to obtain approximate posterior genotype probabilities for pedigree members. The conditional probability that pedigree member i has genotype u_i, given all the phenotypic data $\mathbf{y}$ on the pedigree, is

$$Pr(u_i|\mathbf{y}) = [a_i(u_i)g(y_i|u_i)\prod_{j\in S_i} p_{i,j}(u_i)]/L.$$



$p_{1,2}(u_1)$	$a_2(u_2)$		
	$p_{2,3}(u_2)$	$a_3(u_3)$	
		$p_{5,4}(u_5)$	$\mathbf{a_4(u_4)}$
	$\mathbf{p_{4,5}(u_4)}$		

Fig. 4 A Pedigree used to illustrate modifications to the recursive process to compute posterior probabilities when the posterior family of an offspring has been visited. The direction of the start of the recursive process is indicated by an *arrow*. **B** Probabilities required to recursively calculate $p_{1,2}(u_1)$ in the pedigree of part A. Each column gives the probabilities required to calculate those in the column to the left of it. The recursive process to compute the probabilities printed in *bold-face* are not initiated. **C** Cut pedigree resulting from the application of the modified algorithm to the pedigree in part A

Calculation of genotype probabilities using the modified algorithm is equivalent to cutting all loops in the pedigree and obtaining genotype probabilities from the cut pedigree. However, applying the modified algorithm to compute genotype probabilities for different individuals may result in cutting the loops at different places. Therefore, anterior and posterior probabilities that have been computed for one individual cannot be reused in computing the genotype probabilities for another individual, as in the FSE algorithm for pedigrees without loops. An efficient way to compute genotype probabilities for several pedigree members is to use the modified algorithm to cut the pedigree, and then apply the FSE algorithm to the cut pedigree. Thus, anterior and posterior probabilities computed for one individual can be reused, where necessary, in computing genotype probabilities for another individual. This strategy was used by Thompson (1981).

Numerical examples

As a first example, consider the large pedigree of chickens that was used as a control line in a selection experiment of Hagger (1992). The phenotype ‘egg weight’ is observed on 3276 individuals of a large pedigree comprising seven generations. The pedigree contains several loops, but due to the size of the pedigree, it was not possible to determine visually the location of the loops in the pedigree. The program packages SALP (Stricker et al. 1995), PAP (Hasstedt 1994) and MEN-

DEL (Lange 1991) were used to try and compute the likelihood of a model with a single co-dominant locus with two alleles. PAP and MENDEL calculate the exact likelihood for pedigrees with loops, whereas SALP uses the algorithm presented here to approximate the likelihood. With the programs PAP and MENDEL, it was not possible to compute the exact likelihood on a SUN SPARCstation 10 with 96 megabytes of memory. Although PAP is able to determine a peeling sequence automatically, even in pedigrees with loops, the resulting cutset size of 181 individuals could not be handled computationally. To determine a better peeling sequence would require one to inspect the pedigree visually which is not feasible either. MENDEL, on the other hand, could only handle less than 200 individuals in a pedigree on our computer. However, with the algorithm presented here, SALP was able to approximate the likelihood by introducing 489 additional founders in the pedigree, i.e., cutting 489 loops. The CPU time to compute a single likelihood was less than a second.

As a second example, the modified pedigree structure from Strickberger (1976), p 235, and shown in Fig. 5, was analyzed. The pedigree consists of 72 individuals; three additional dummy individuals with a missing phenotype had to be introduced as parents to fullfill the requirement that each individual has either both parents or no parent in the pedigree. Under the assumption of a single locus with two alleles, say *a* and *A*, a valid Mendelian but arbitrary genotypic constellation for the pedigree was established. Depending on the genotype of an individual (*aa*, *aA* or *AA*) a random number from a distribution $N(40, 25)$, $N(50, 25)$ or $N(60, 25)$ was assigned as the individuals phenotype, respectively. There are four loops in the pedigree and they can be identified by visual inspection. PAP was used to compute the exact maximum likelihood for a single co-dominant locus with two alleles on this looped pedigree. SALP was applied to the data to compute the approximate likelihood for the same model by loop-cutting. The four loops were cut, introducing additional founders for individuals 8, 9, 11 and 67. The maximum likelihoods and corresponding maximum-likelihood estimates for PAP and SALP are shown in Table 1. Note that the maximum likelihood and corresponding estimates from PAP and SALP in fact refer to different pedigrees. The results obtained by SALP are exactly the same as those obtained when the cut pedigree (with the four additionally introduced founders) is analyzed by PAP. Subsequently, genotype probabilities of all members of the looped and the cut pedigree were computed using PAP and SALP, respectively. The distribution of the differences between the genotype probabilities, computed on the looped (PAP) and cut (SALP) pedigree, are shown in Fig. 6 for the homozygous genotype *aa*, in Fig. 7 for the heterozygous genotype *aA*, and in Fig. 8 for the homozygous genotype *AA*, respectively.

Discussion

The primary goal of this paper was to present an algorithm to cut arbitrary large complex pedigrees with

Table 1 Maximum likelihood and corresponding estimates for the looped pedigree (obtained by PAP) and the cut pedigree (obtained by SALP)

	PAP	SALP
ln likelihood	−250.6744	−264.0294
Gene frequency	0.31	0.37
Difference between homozygous genotypic means	17.16	16.45
Difference between genotypic mean *aa* and heterozygous genotypic mean	14.37	14.45
Residual variance	5.37	5.28

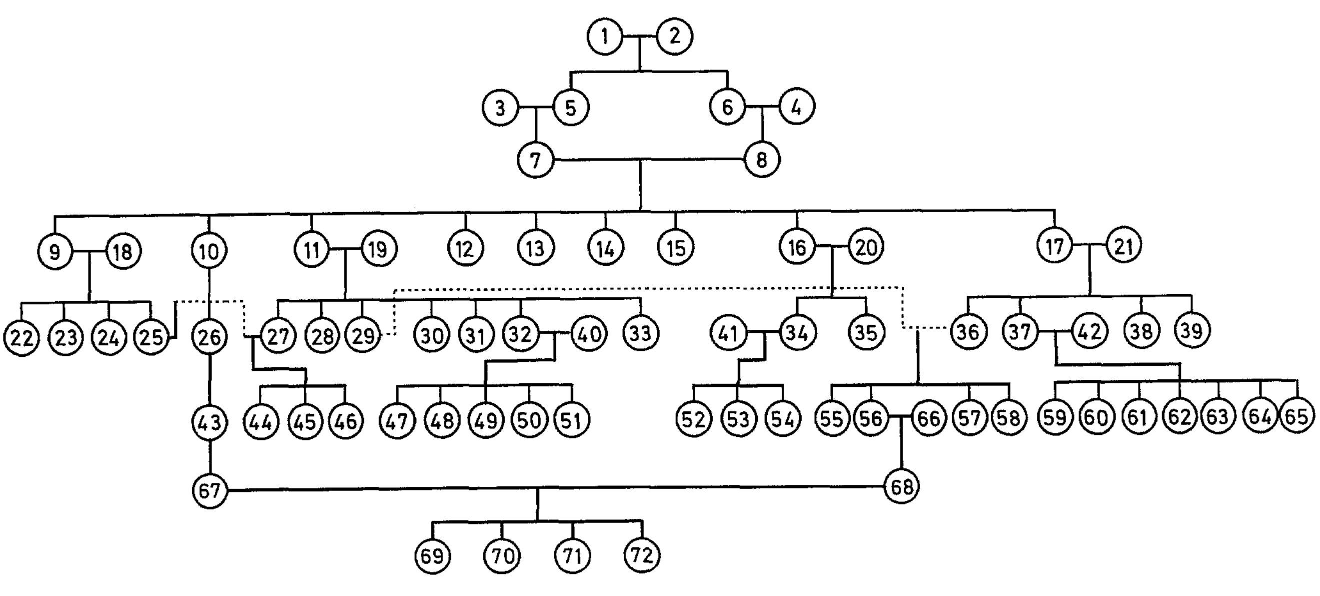


Fig. 5 Original 72-member pedigree from Strickberger (1981) containing four consanguineous loops

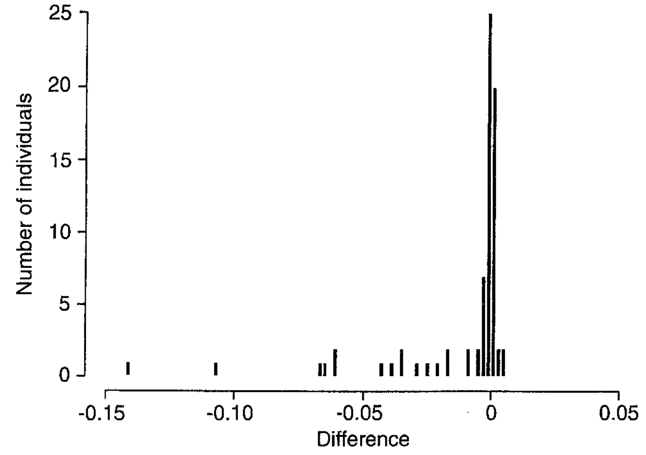


Fig. 6 Distribution of the difference between genotype probabilities for the homozygous genotype *aa* computed on the looped pedigree (obtained by PAP) and on the cut pedigree (obtained by SALP) for all 75 individuals in the pedigree

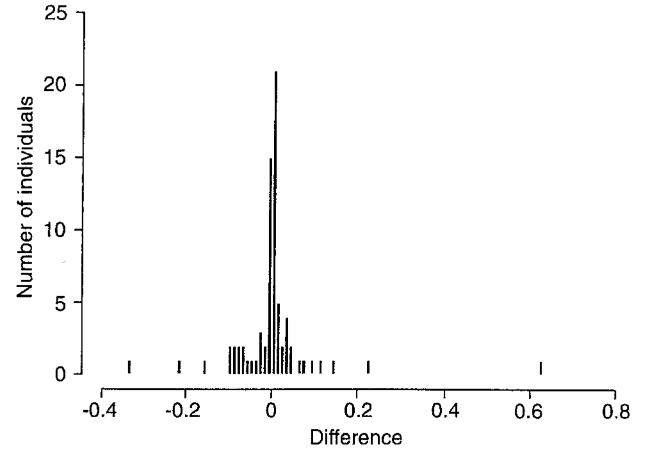


Fig. 7 Distribution of the difference between genotype probabilities for the heterozygous genotype *aA* computed on the looped pedigree (obtained by PAP) and on the cut pedigree (obtained by SALP) for all 75 individuals in the pedigree

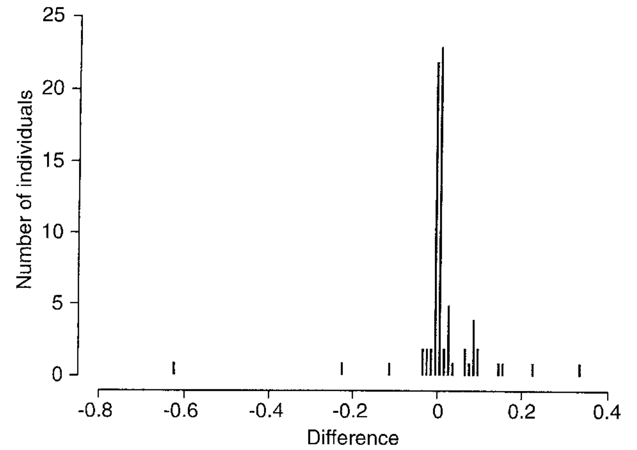


Fig. 8 Distribution of the difference between genotype probabilities for the homozygous genotype *AA* computed on the looped pedigree (obtained by PAP) and on the cut pedigree (obtained by SALP) for all 75 individuals in the pedigree

loops and simultaneously compute the likelihood without the need of either visually inspecting the pedigree or using a method that may require excessive computation time or storage space. Cutting a loop is performed by introducing an artificial founder individual into the pedigree and assigning it population genotype proportions as anterior probabilities and the same phenotype as the original individual that was duplicated. Performing cuts in a pedigree with loops according to the algorithm presented here is equivalent to the way pedigrees with loops are usually handled, when the loops can be identified by visual inspection. Applying the algorithm will exactly automate what has been done in human genetics previously (e.g., Amos et al. 1986) and thus this algorithm will allow one to approximate the likelihood for pedigrees too large in size to identify loops visually.

The first numerical example illustrates this fact clearly. Whereas exact analysis of this 3276-member pedigree with 489 loops is not feasible using PAP or MENDEL, the algorithm presented here allowed an approximation of the likelihood through a cut pedigree.

Some aspects of the accuracy of the approximation are addressed by the second numerical example. The difference between the log likelihoods can obviously not be interpreted, since they are based on different pedigrees. PAP computes the likelihood for the looped pedigree, SALP for a cut pedigree with additional members. However, even for this relatively small pedigree where the amount of information artifically introduced and that ignored because of dependencies among individuals in a loop is not negligable, the maximum-likelihood estimates for the looped (PAP) and cut (SALP) pedigree shown in Table 1 are similar. Overall, the genotype probabilities obtained from the looped pedigree and the cut pedigree were in good agreement, but substantial differences were observed for a few individuals Figs. (6–8). The genotype probabilities computed for the individuals where a cut is performed, and for individuals closely connected to them by marriage or relationship, will be biased most. The bias will increase with increasing dependency, i.e., with decreasing size of the loop that was cut. All loops present in this pedigree were relatively small or nested, indicating that a considerable amount of information about relationship was ignored when the loops were cut. The largest difference in absolute value between genotype probabilities ($>|0.2|$) occured at individuals 6, 8, 18 and 67, all of which are part of loops and are either individuals where a loop was cut or the spouse or parent of such an individual. The algorithm automatically identifies the individuals which are duplicated when a loop is cut. Using this information, genotype probabilities for individuals in the area where a loop is cut may be interpreted with caution. This small example is not meant to give a complete indication of the accuracy of the approximation to compute genotype probabilities. For example, it ignores the fact that applying the algorithm to the pedigree with another peeling sequence would result in a different cut pedigree. However, it shows that cutting small loops, i.e., ignoring a relatively strong dependency between individuals, may result in poor estimates for genotype probabilities of individuals in the area where a loop was cut and that such genotype probabilities should be interpreted with caution. Following Lange and Elston (1975), it should be noted that if the duplicated individuals have identical phenotypes and their genotype can be unequivocally inferred from their phenotype, the approximation is exact. Based on this, it would be possible to generate examples where cutting a loop by the algorithm presented here would be exact, or, on the other hand, lead to a worse approximation.

The strength of this algorithm lies in its ability to compute an approximate likelihood in large and complex pedigrees where the exact likelihood cannot be calculated. The accuracy of the approximation is affected by the number, size, and nesting level of the loops. For large and complex pedigrees containing only a few and large loops, the algorithm enables approximate genetic analysis where exact algorithms may fail.

Appendix

Definition of a loop in a pedigree

A pedigree can be considered to be a finite, connected graph, i.e., a set of vertices or points together with a set of edges. Two types of vertices exist in a pedigree, vertices representing individuals and vertices representing marriages. Edges occur only between individual and marriage vertices and thus connect individuals with a marriage vertex. Fig. 9 illustrates these definitions. A pedigree is said to have a loop if the corresponding graph contains a cycle.

Let n_p be the number be the number of parents in a pedigree, i.e., the number of individuals who have offspring in the pedigree; let n_o be the number of offspring in the pedigree, i.e., the number of individuals who have both parents in the pedigree; let n_m be the number of mating pairs in that pedigree; and let n_{po} be the number of pedigree members who are both parents and offspring, i.e., the number of individuals who have parents and at least one offspring in the pedigree. Then the pedigree contains a loop if, and only if, $n_m \geqslant n_p - n_{po} - 1$.

Proof: referring to Bondy and Murty (1976, p 25), a finite connected graph contains no cycles if, and only if, the number of edges equals the number of vertices minus 1; and it contains at least one cycle if the number of edges is larger than this. The number of edges in the graph of a pedigree is equal to $2n_m + n_o$. The number of vertices is the number of marriage vertices plus the number of individual vertices., i.e., $n_m + n_p + n_0 - n_{po}$. Thus, if

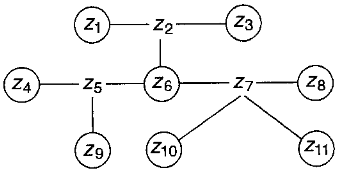
$$2n_m + n_o = n_m + n_p + n_o - n_{po} - 1$$

the pedigree contains no loop; and if

$$n_m \geqslant n_p - n_{po} - 1$$

the pedigree contains a loop.

Fig. 9 Example of a graph theoretical representation of a pedigree in which the individual vertices are *circled*. Individual z_1 is mated to z_3, whose offspring z_6 is mated to z_4 and z_8, producing offspring z_9, z_{10} and z_{11}. The corresponding marriage vertices are z_2, z_5 and z_7. It can be seen that edges occur only between individual and marriage vertices



Acknowledgements This work was supported by the Schweizerischer Nationalfonds, Switzerland (C. S.), by the Illinois Agricultural Experiment Station, Hatch Project 35-0345 (R. L. F.) and by the U. S. Public Health Service research grant GM 28396 from the National Institute of General Medical Sciences and resource grant 1P41PR03655 from the National Center for Research Resources (R. C. E.).

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

Bondy JA, Murty USR (1976) Graph theory with applications. North Holland, New York

Cannings C, Thompson EA, Skolnick MH (1978) Probability functions on complex pedigrees. Adv Appl Prob 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

Hagger C (1992) Two generations of selection on restricted best linear unbiased prediction breeding values for income minus feed cost in laying hens. J Anim Sci 70:2045–2052

Hasstedt SJ (1994) PAP: pedigree analysis package, rev 4. Department of Human Genetics, University of Utah, Salt Lake City

Janss LLG, Van der Werf JHJ, Van Arendonk JAM (1992) Detection of a major gene using segregation analysis in data from several generations. In: Proc Europ Assoc Anim Prod, Session 5a, Free Communications, p 144

Kong A (1991) Analysis of pedigree data using methods combining peeling and gibbs sampling. In: Keramidas EM (ed) Computing science and statistics. Proc 23rd Symp on the Interface, April 21–24, pp 379–384

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

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

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

Sheehan NA (1989) Image-processing procedures applied to the estimation of genotypes on pedigrees. Technical Report 176, Department of Statistics, University of Washington, Seattle, Washington, USA

Strickberger MW (1976) Genetics MacMillan Publishing Company, New York

Stricker C, Fernando RL, Elston RC (1995) SALP—Segregation and linkage analysis for pedigrees, release 2.0, computer program package. Federal Institute of Technology ETH, Institute of Animal Sciences, Zürich, Switzerland

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

Thompson EA (1981) Pedigree analysis of hodgkins disease in a newfoundland genealogy. Ann Hum Genet 45:279–292

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

Thompson EA, Wijsman EM (1990) The gibbs sampler on extended pedigrees: Monte carlo method for the genetic analysis of complex traits. Technical Report 193, Department of Statistics, GN-22, University of Washington, Seattle, Washington, USA