An efficient algorithm to compute marginal posterior genotype probabilities for every member of a pedigree with loops

Liviu R Totir¹, Rohan L Fernando*² and Joseph Abraham³

Address: ¹Pioneer Hi-Bred International, A Dupont Business, 7250 NW 62nd Ave, Johnston, Iowa 5013, USA, ²Department of Animal Science and Center for Integrated Animal Genomics, Iowa State University, Ames, Iowa 50011, USA and ³Case Western Reserve University, Cleveland, Ohio 44106, USA

Email: Liviu R Totir - radu.totir@pioneer.com; Rohan L Fernando* - rohan@iastate.edu; Joseph Abraham - jabraham@darwin.EPBL.cwru.edu

* Corresponding author

Abstract

Background: Marginal posterior genotype probabilities need to be computed for genetic analyses such as genetic counseling in humans and selective breeding in animal and plant species.

Methods: In this paper, we describe a peeling based, deterministic, exact algorithm to compute efficiently genotype probabilities for every member of a pedigree with loops without recourse to junction-tree methods from graph theory. The efficiency in computing the likelihood by peeling comes from storing intermediate results in multidimensional tables called cutsets. Computing marginal genotype probabilities for individual \(i\) requires recomputing the likelihood for each of the possible genotypes of individual \(i\). This can be done efficiently by storing intermediate results in two types of cutsets called anterior and posterior cutsets and reusing these intermediate results to compute the likelihood.

Examples: A small example is used to illustrate the theoretical concepts discussed in this paper, and marginal genotype probabilities are computed at a monogenic disease locus for every member in a real cattle pedigree.

Background

For monogenic or oligogenic traits, algorithms for efficient likelihood computations have been described for both pedigrees without loops [1], and pedigrees with loops [2-5] Furthermore, efficient algorithms have been developed to draw samples from the joint posterior distribution of genotypes from complex pedigrees [6,7]. However, when pedigrees are large with many loops and multiple loci, these sampling methods can become very inefficient, and the J-PCS algorithm was proposed to address this problem [8]. This algorithm involves a) modifying the pedigree by cutting some loops and b) sampling the genotype of an individual \(i\) that is as distant as possible from the modifications ("cuts"). This sample must be drawn from the marginal posterior genotype probability distribution of \(i\) given the modified pedigree, which may still have many loops. Furthermore, marginal posterior genotype probabilities are needed in genetic counseling in humans and selective breeding in domesticated species. An efficient, exact, deterministic algorithm is available to compute the marginal posterior genotype probabilities for every member in a pedigree without loops [9]. However, it is not straightforward how to extend this algorithm to compute marginal posterior genotype probabilities for pedigrees with loops. Recently, junction tree methods from graph theory were used to describe an efficient algo-
rithm to compute marginal posterior genotype probabilities for pedigrees with loops [10]. Most geneticists, however, are not familiar with junction tree concepts, and thus such algorithms would not readily be incorporated in genetic analyses, especially because the paper of Lauritzen and Sheehan [10] is not self-contained, but relies on results from other sources. In this paper, we present a self-contained description of an efficient, exact, deterministic algorithm to compute marginal posterior genotype probabilities for every member of a pedigree with loops, without use of junction tree methods. This algorithm has been implemented in the public domain software package MATVEC and can be obtained from the corresponding author.

Following is a brief outline of the presentation. First we define pedigree loops. Next we discuss the relationship between the likelihood and marginal posterior genotype probabilities of pedigree members. Following this, anterior and posterior cutsets are introduced. Anterior cutsets are used to compute the likelihood by the Elston-Stewart algorithm (peeling), and anterior and posterior cutsets are used to describe an efficient algorithm to calculate marginal probabilities for every member of a pedigree with loops. Next, marginal genotype probabilities are calculated for every member in a cattle pedigree that contains loops. Finally, in the appendix, a small example is used to illustrate in detail the theoretical concepts discussed in this article.

**Methods**

**Definition of Pedigree Loops**

Here we define pedigree loops indirectly by providing a simple algorithm to determine if a pedigree contains loops. A pedigree is a set of individuals, each of which can be classified as a founder or a non-founder. A founder is a pedigree member whose parents are not in the pedigree, and a non-founder is a pedigree member with both parents present in the pedigree. A nuclear family consists of a set of parents and all their offspring. A terminal family is a family that has at most one member who belongs to at least one other nuclear family. Terminal members of a pedigree are members of terminal families that do not belong to another family. The algorithm used to determine if a pedigree contains loops relies on identifying and then eliminating terminal members from the pedigree. If a pedigree does not contain any loops, repeated removal of terminal members from the pedigree will result in all members being removed from the pedigree. On the other hand, if a pedigree contains any loops, not all members of the pedigree can be removed by repeated removal of terminal members. See additional file 1: "Algorithm to detect loops.pdf" for an example of the use of this algorithm to identify loops in arbitrary pedigrees.

**Likelihood and Genotype Probability Calculations for General Pedigrees**

Consider a pedigree with \( n \) individuals, and let \( g_i \) denote the possible genotype and \( y_i \) the observed phenotype of an arbitrary pedigree member \( i \). Note that both \( g_i \) and \( y_i \) can be a function of a single locus or of multiple loci on the chromosome. The likelihood for a genetic model given the observed data can be written as

\[
L(\rho, q, \theta; y) = \sum_g F(g, y; \rho, q, \theta)
\]  

where \( F(g, y; \rho, q, \theta) \) denotes the joint distribution of all \( g_i \) (\( g \)) and all \( y_i \) (\( y \)) in the pedigree, \( \rho \) is the vector of recombination rates between loci, \( q \) is the vector of gene frequencies, and \( \theta \) is the vector of parameters in the genetic model that relates \( y_i \) and \( g_i \) [11]. Furthermore, the likelihood can be written as

\[
L(\rho, q, \theta; y) = \sum_{s_i} \ldots \sum_{s_n} f_i(g_{s_i}) \ldots f_n(g_{s_n}),
\]

where \( g_{s_i} \) is a set of possible genotypes of a given set of pedigree members \( s_i \), and \( f_i(g_{s_i}) \) is defined as

\[
f_i(g_{s_i}) = \begin{cases} 
    h(y_i | g_i, \theta) \times \Pr(g_i | q) & \text{for } s_i = \{i\}, \\
    h(y_i | g_i, \theta) \times \Pr(g_i | g_{m_i}, g_{f_i}, \rho) & \text{for } s_i = \{i, m_i, f_i\},
\end{cases}
\]

where \( h(y_i | g_i, \theta) \) is the conditional probability of the phenotype \( y_i \) given the genotype \( g_i \), \( \Pr(g_i | q) \) is the marginal probability that a founder has genotype \( g_i \), \( \Pr(g_i | g_{m_i}, g_{f_i}, \rho) \) is the probability that a non-founder has genotype \( g_i \) given that its mother \( m_i \) has genotype \( g_{m_i} \) and its father \( f_i \) has genotype \( g_{f_i} \), (transition probability). When \( g_{m_i} \) and \( g_{f_i} \) consist of multiple loci, the multilocus transition probability can be written as a product of single-locus transition probabilities and recombination probabilities between adjacent loci, by making use of the Markov property for recombination events between adjacent loci that holds under the assumption of no interference [5,12]. Note that, for each individual \( i \) in the pedigree, a set \( s_i \) is defined that contains either one or three individuals. For founders, \( s_i \) contains only \( i \), while for non-founders, \( s_i \) contains \( i, m_i \) and \( f_i \). For an arbitrary pedigree member \( i \), marginal genotype probabilities can be written as
\[ \Pr(g_i = x) = \frac{L_{g_i=x}}{L}, \]  

where \( L \) is the likelihood defined in 2, and \( L_{g_i=x} \) is the likelihood computed with \( g_i \) fixed at genotype \( x \). Thus, the efficient computation of marginal genotype probabilities using equation 4 requires an efficient algorithm to compute the likelihood. The computation of the likelihood using 2 is not efficient for pedigrees having more than about 20 members. However, the Elston-Stewart algorithm, which is also known as peeling, can be used to efficiently compute the likelihood \([1,13]\). Still, using equation 4 to compute marginal probabilities for \( N \) unknown genotypes of individual \( i \) requires recompute the likelihood with \( g_i = x \) for each of the \( N \) values of \( x \). Furthermore, this has to be repeated for all \( n \) individuals in the pedigree. In the following section we introduce an algorithm to avoid repeating computations by storing intermediate results in multidimensional tables called anterior and posterior cutsets.

**Anterior and Posterior Cutsets**

Computing the likelihood by peeling involves summing over the genotypes of one individual at a time and storing the intermediate results. For convenience, here we assume that individuals are numbered in the order that they are peeled. Peeling the first individual amounts to computing the sum over \( g_1 \) of the product of all factors in 2 that contain \( g_1 \), for each combination of the other genotypes that occur together with \( g_1 \). Results of these summations are stored in a multidimensional table that has been called a cutset \([13]\). Here we will refer to these tables as anterior cutsets. The anterior cutset obtained after peeling \( g_1 \) will be denoted by \( C^A_i(g_{V_1}) \) and is calculated as

\[ C^A_i(g_{V_1}) = \sum_{s_1} \prod_{j} f_j(g_{s_j}), \]  

where \( V_1 \) is a set of pedigree members defined as follows. Using the sets \( s_i \) defined earlier for each individual in the pedigree, \( U_1 \) is defined as the union of all \( s_i \) that contain individual 1. Then \( V_1 \) is obtained by removing individual 1 from \( U_1 \). Further, \( g_{V_1} \) is the set of genotypes for the individuals in \( V_1 \). Note that the product in 5 is over those pedigree members \( j \) that contain individual 1 in their \( s_j \).

Replacing in 2 the product of all factors containing \( g_i \), summed over \( g_i \) with \( C^A_i(g_{V_1}) \) gives the following expression for the likelihood

\[ L = \sum_{s_1} \prod_{r} f_r(g_{s_r}) C^A_i(g_{V_1}) \]  

where \( g_i = \{g_2 \ldots g_n\} \) is the set of possible genotypes of the individuals that remain to be peeled, and the product is over those pedigree members \( r \) that do not contain individual 1 in their \( s_r \). The likelihood expressed as above after peeling \( g_i \), will be referred to as \( LE_i \), and in general after peeling \( g_n \) will be referred as \( LE \).

Note that after \( g_1 \) has been peeled, the summation in 6 is only over the genotypes of individuals 2 ... \( n \). As described below, and later illustrated through a hypothetical example in the Appendix, as each individual is peeled, an anterior cutset is generated. After peeling the last individual, the final anterior cutset will have only a single value that is equal to the likelihood. Note that for a pedigree with \( n \) members, there are \( n! \) possible peeling orders. Although any choice of a peeling sequence will lead to the same value for the likelihood, not all choices of the peeling sequence lead to anterior cutsets of the same size. As the amount of memory required does depend on the size of the cutsets, a peeling sequence leading to smaller cutsets is more desirable. However, even for moderately large \( n \), an exhaustive search for an efficient peeling sequence is not feasible. Furthermore, there is no known algorithm to efficiently find the peeling order with the lowest storage requirements \([10]\). However, the following simple heuristic procedure can be used to generate a good peeling sequence. At any stage of the peeling process, in order to decide which individual should be peeled next, for each individual \( i \) that remains to be peeled, we compute the size of the anterior cutset that would be generated by peeling \( i \). The individual with the smallest anterior cutset size is chosen to be peeled next \([14]\).

Now it is convenient to introduce the posterior cutset which will be used to avoid repeating computations in calculating genotype probabilities. By factoring out \( C^A_i(g_{V_1}) \) from 6 and by summing over the genotypes of all remaining pedigree members not contained in \( V_1 \), we can define a second multidimensional table called a posterior cutset

\[ C^P_i(g_{V_1}) = \sum_{s_1 \in g_{V_1}} \prod_{r} f_r(g_{s_r}) \]  

(page number not for citation purposes)
where \( C^P_i(g_{V_i}) \) is not a function of \( g_i \). As a result we can rewrite the likelihood as follows

\[
L = \sum_{g_{V_i}} C^A_i(g_{V_i}) C^P_i(g_{V_i}).
\]  

(8)

In the general description of peeling given below, we make extensive use of two sets defined for each individual \( i \). The first set \( s_i \) has already been described earlier, and it is completely determined by the pedigree. The second set \( V_i \) contains the individuals in the cutset that is generated when \( i \) is peeled. Thus, \( V_i \) is determined by the pedigree and the peeling order. In general, peeling individual \( i \) amounts to computing the sum over \( g_i \) of the product of all factors in \( LE_i \) that contain \( g_i \) for each combination of the other genotypes that occur together with \( g_i \). These summations are stored in the anterior cutset for \( i \):

\[
C^A_i(g_{V_i}) = \sum_{g_i} \prod_j f_j(g_{s_j}) \prod_k C^A_k(g_{V_k}).
\]

(9)

where \( j \) is an individual whose function \( f_j(g_{s_j}) \) remains in \( LE_{j-1} \) and \( i \) is an individual whose anterior cutset \( C^A_k(g_{V_k}) \) remains in \( LE_{i-1} \) and \( i \) is in \( V_k \), \( U_i = (\cup_{s_j} \cup V_k) \), and \( V_i = U_i \setminus i. \) Replacing in \( LE_{i-1} \) the sum over \( g_i \) of the product of all factors containing \( g_i \) with \( C^A_i(g_{V_i}) \) gives the likelihood expression \( LE_i \):

\[
L = \sum_{g_i} \prod_j f_j(g_{s_j}) \prod_k C^A_k(g_{V_k}) C^P_i(g_{V_i}).
\]

(10)

where \( f_j(g_{s_j}) \) are the functions from \( LE_{i-1} \) that were not used in the calculation of \( C^A_i(g_{V_i}) \) and \( C^A_k(g_{V_k}) \) are the anterior cutsets from \( LE_{i-1} \) that were not used in the calculation of \( C^A_i(g_{V_i}) \). Now we obtain the posterior cutset for \( i \) by removing \( C^A_i(g_{V_i}) \) from \( LE_i \):

\[
C^P_i(g_{V_i}) = \sum_{g_i \notin g_{V_i}} \prod_j f_j(g_{s_j}) \prod_k C^A_k(g_{V_k}).
\]

(11)

Note that \( C^P_i(g_{V_i}) \) is not a function of \( g_i \). Thus, in general we can write the likelihood as follows

\[
L = \sum_{g_{V_i}} C^A_i(g_{V_i}) C^P_i(g_{V_i}).
\]

(12)

Now we are ready to explain how to compute genotype probabilities for any individual \( m \in V_i \) using anterior and posterior cutsets. As in equation 4, marginal genotype probabilities for \( m \) can be written as

\[
Pr(g_m=x) = \frac{L \cdot g_m=x}{L}.
\]

(13)

The denominator of 13 is given by 12, while the numerator is obtained by computing 12 with \( g_m \) fixed at \( x. \) If \( m \) is in more than one set of pedigree members \( V_i \) identifying the set \( V_i \) with smallest number of members will minimize the required computations. However, if \( m \) is not in any \( V_i \) we first write the likelihood 12 as a product of the anterior and posterior cutsets for \( m. \) In this expression, however, \( m \) has already been peeled. Equation 9, which is used to compute the anterior cutset for an arbitrary individual, contains that individual prior to it being peeled. Thus, by substituting in 12, the expression given in 9 for \( C^A_m(g_{V_m}) \) gives

\[
L = \sum_{g_{V_m}} \sum_{g_m} \prod_j f_j(g_{s_j}) \prod_k C^A_k(g_{V_k}) C^P_m(g_{V_m}).
\]

(14)

Now the numerator of 13 is obtained by computing 14 with \( g_m \) fixed at \( x. \)

Provided a good peeling sequence is available, computation of the required anterior cutsets and the summation over \( g_{V_i} \) in 12 or \( g_{V_m} \) in 14 would be feasible. However, posterior cutsets cannot be computed efficiently using 11 because here the summation may be over a very large set of genotypes. Fortunately, posterior cutsets can be computed recursively as described below. Although the derivation of the recursive algorithm given below is conceptually straightforward, it may be tedious to follow. Thus, at the end of this section, we provide four easy to implement steps to efficiently compute posterior cutsets.

The key principle that we have used to compute marginal posterior probabilities efficiently is that any pedigree member can be assigned into one of three mutually exclusive sets with respect to any individual \( i: \) the set of members that contribute to \( C^A_i(g_{V_i}) \), the set of members that contribute to \( C^P_i(g_{V_i}) \), or the set of members in \( V_i \). For example, in computing the numerator of 13 by using 12, the intermediate results from peeling individuals in the
first set were stored in $C_i^A(g_{V_i})$ and used repeatedly, the intermediate results from peeling individuals in the second set were stored in $C_i^P(g_{V_i})$ and used repeatedly, and only the calculations for peeling individuals in the third set were repeated. This principle of factoring the likelihood into anterior and posterior components is used repeatedly in the following derivations. To derive the recursive algorithm, first we establish that $C_n^P() = 1.0$, which is the base case of the recursion. Similar to 10, after peeling individual $n - 1$, the likelihood expression $LE_{n-1}$ becomes

$$L = \sum_{g_n} f_n(g_n) \prod_u C_u^A(g_{V_u}) C_{n-1}^A(g_{V_{n-1}}).$$

(15)

Because only individual $n$ remains to be peeled, $V_n$ and $V_{n-1}$ contain only $n$. The likelihood now becomes

$$L = \sum_{g_n} f_n(g_n) \prod_u C_u^A(g_{V_u}) C_{n-1}^A(g_{V_{n-1}}).$$

(16)

Further, using 9, $C_n^A()$ can be written as

$$C_n^A() = \sum_{g_n} f_n(g_n) \prod_u C_u^A(g_{V_u}) C_{n-1}^A(g_{V_{n-1}}).$$

(17)

Note that in 16 and 17 the right-hand sides are identical, and thus $L = C_n^A()$. However, from 12

$$L = C_n^A() C_n^P().$$

(18)

and thus $C_n^P() = 1.0$. Now, for any other individual $i$, $C_i^P(g_{V_i})$ can be computed recursively as follows.

The anterior cutset $C_i^A(g_{V_i})$ generated when $i$ is peeled, is used in the calculation of the anterior cutset generated when $k = \min(V_i)$ is peeled. The resulting anterior cutset can be written as

$$C_k^A(g_{V_i}) = \sum_{g_k} \prod_r f_r(g_s) \prod_j C_j^A(g_{V_j}) C_i^A(g_{V_i})$$

(19)

where $f_r(g_s)$ are all remaining functions with $k \in s$, and $C_j^A(g_{V_j})$ are the remaining anterior cutsets with $k \in V_j$ in addition to $C_i^A(g_{V_i})$. Similar to (12) we can also write

$$L = \sum_{g_k} C_k^A(g_{V_k}) C_i^P(g_{V_i}).$$

(20)

and by using (19) in (20) we can write

$$L = \sum_{g_k} \sum_{g_i} \prod_r f_r(g_s) \prod_j C_j^A(g_{V_j}) C_i^A(g_{V_i}) C_k^P(g_{V_k}).$$

(21)

Recall that we have defined the set of individuals $U_k = V_k \cup \{k\}$, and thus we can write

$$L = \sum_{g_k} \prod_r f_r(g_s) \prod_j C_j^A(g_{V_j}) C_i^A(g_{V_i}) C_k^P(g_{V_k}).$$

(22)

Note that both (12) and (22) contain the term $C_i^A(g_{V_i})$. By rearranging 22, the likelihood can be written as

$$L = \sum_{g_k} \sum_{s_{ik} \in g_{V_i}} \prod_r f_r(g_s) \prod_j C_j^A(g_{V_j}) C_i^A(g_{V_i}) C_k^P(g_{V_k}),$$

(23)

and using 12 we can write

$$C_i^P(g_{V_i}) = \sum_{s_{ik} \in g_{V_i}} \prod_r f_r(g_s) \prod_j C_j^A(g_{V_j}) C_k^P(g_{V_k}).$$

(24)

Thus, the posterior cutset for individual $i$ can be expressed as a function of some anterior cutsets and the posterior cutset for individual $k > i$. Starting at individual $n - 1$ all posterior cutsets can be computed in the reverse order of peeling because $C_n^P() = 1.0$.

In summary, the following procedure can be used to recursively compute the posterior cutset of an arbitrary individual $i$ in a pedigree:

1. Compute anterior cutsets for all individuals in the pedigree. This step is done only once.

2. Identify the anterior cutset $C_i^A(g_{V_i})$ whose summand contains the factor $C_i^A(g_{V_i})$ (see equation 19).
3. Replace $C_i^A(g_{V_i})$ in the summand of $C_k^A(g_{V_k})$ with $C_k^P(g_{V_k})$, and for each value of $g_{V_i}$ sum over the remaining genotypes in this expression (see equation 24).

4. If $C_k^P(g_{V_k})$ has not been computed yet, use steps 2, 3 and 4 to compute it (this is the recursion).

Note that to compute marginal posterior genotype probabilities for an arbitrary member of the pedigree using this algorithm, we need to calculate all anterior cutsets and a subset of all posterior cutsets. Both the anterior and the posterior cutset of a given individual have the same size. The computation of an anterior cutset involves the summation over the genotypes of one individual. The computation of a posterior cutset can involve summations over the genotypes of a variable number individuals. The theoretical concepts introduced in this section are illustrated in detail for a simple example in the Appendix. In the following section we discuss a real data application of the theoretical concepts described above.

**Genotype Probabilities Computations in a Real Cattle Pedigree**

Consider the pedigree given in the first three columns of Table 1 with a graphical representation given in Figure 1. Six terminal members of this cattle pedigree (individuals A21, A22, A23, A24, A25 and A26) are known to be affected by a monogenic recessive disease. Conditional on disease status, assumed mode of inheritance, pedigree information, and on the assumption that the frequency of the recessive allele in the cattle population from which the pedigree was sampled is equal to 0.00001, we calculate genotype probabilities for every member of the pedigree using the algorithm described above. Of the six founders present in this cattle pedigree, founder individual A2 is identified to be a carrier of the recessive allele with probability 1.0. Selective breeding decisions can be made given the calculated posterior genotype probabilities.

Next, we augment the genetic information used to calculate posterior genotype probabilities, by including genetic data on two marker loci flanking the hypothesized position of the recessive locus. Each marker locus has three alleles and the two loci are separated by 0.8 cM with the hypothesized position of the recessive locus 0.5 cM from the left marker (M1). The allele scores of the two markers used are given in Table 2. The impact of the additional information provided by the marker data is reflected in the posterior probability of individuals A19 and A20 being carriers of the recessive allele (Table 3). While without marker data individuals A19 and A20 have a posterior probability of being carriers equal to 0.6667, with marker data the probability is close to one.

### Table 1: Genetic profile of 26 individuals conditional on pedigree and phenotypic data.

| Individual | Dam | Sire | Phenotype | Genotype Probabilities |
|------------|-----|------|-----------|------------------------|
| A1, A4, A6 | 0   | 0    | Normal    | Pr(0/0) 0.99999, Pr(0/1) 0.000005, Pr(1/0) 0.000005, Pr(1/1) 0.0 |
| A2         | 0   | 0    | Normal    | Pr(0/0) 0.0, Pr(0/1) 0.5, Pr(1/0) 0.5, Pr(1/1) 0.0 |
| A3, A5     | 0   | 0    | Normal    | Pr(0/0) 1.0, Pr(0/1) 0.0, Pr(1/0) 0.0, Pr(1/1) 0.0 |
| A7         | A1  | A2   | Normal    | Pr(0/0) 0.0, Pr(0/1) 1.0, Pr(1/0) 0.0, Pr(1/1) 0.0 |
| A8         | A3  | A2   | Normal    | Pr(0/0) 0.00001, Pr(0/1) 0.99999, Pr(1/0) 0.0, Pr(1/1) 0.0 |
| A9, A10, A11| A4  | A2   | Normal    | Pr(0/0) 0.0, Pr(0/1) 0.99999, Pr(1/0) 0.00001, Pr(1/1) 0.0 |
| A12, A13   | A4  | A8   | Normal    | Pr(0/0) 0.0, Pr(0/1) 0.99999, Pr(1/0) 0.00001, Pr(1/1) 0.0 |
| A14        | A5  | A9   | Normal    | Pr(0/0) 0.0, Pr(0/1) 1.0, Pr(1/0) 0.0, Pr(1/1) 0.0 |
| A15, A16   | A6  | A10  | Normal    | Pr(0/0) 0.0, Pr(0/1) 0.99999, Pr(1/0) 0.00001, Pr(1/1) 0.0 |
| A17        | A6  | A10  | Normal    | Pr(0/0) 0.5, Pr(0/1) 0.5, Pr(1/0) 0.0, Pr(1/1) 0.0 |
| A18        | A6  | A11  | Normal    | Pr(0/0) 0.0, Pr(0/1) 0.99999, Pr(1/0) 0.00001, Pr(1/1) 0.0 |
| A19        | A12 | A9   | Normal    | Pr(0/0) 0.33333, Pr(0/1) 0.33333, Pr(1/0) 0.33333, Pr(1/1) 0.0 |
| A20        | A12 | A9   | Normal    | Pr(0/0) 0.33333, Pr(0/1) 0.33333, Pr(1/0) 0.33333, Pr(1/1) 0.0 |
| A21        | A14 | A15  | Affected  | Pr(0/0) 0.0, Pr(0/1) 0.0, Pr(1/0) 0.0, Pr(1/1) 1.0 |
| A22        | A14 | A16  | Affected  | Pr(0/0) 0.0, Pr(0/1) 0.0, Pr(1/0) 0.0, Pr(1/1) 1.0 |
| A23        | A14 | A7   | Affected  | Pr(0/0) 0.0, Pr(0/1) 0.0, Pr(1/0) 0.0, Pr(1/1) 1.0 |
| A24, A25   | A12 | A9   | Affected  | Pr(0/0) 0.0, Pr(0/1) 0.0, Pr(1/0) 0.0, Pr(1/1) 1.0 |
| A26        | A13 | A18  | Affected  | Pr(0/0) 0.0, Pr(0/1) 0.0, Pr(1/0) 0.0, Pr(1/1) 1.0 |

Pr(1/1) denotes the probability of an individual being homozygous for the recessive allele.
Discussion

As stated by Jensen and Kong [15] current algorithms for calculating marginal posterior genotype probabilities by peeling are inefficient. As described earlier, computing marginal genotype probabilities for individual \( j \) using equation 13, requires recomputing the likelihood for each of the possible genotypes of individual \( j \). For the last individual in the peeling sequence, this can be done efficiently because intermediate results from peeling individuals 1 through \( n - 1 \), for each possible value of \( g_n \), have been stored in the anterior cutset \( C_{n-1}^A(g_{n-1}, g_n) \). Thus, by making use of the intermediate results stored in \( C_{n-1}^A(g_n) \), only calculations from the last step of peeling need to be repeated to compute \( L_{g_n=x} \). For any \( m \) that is in more than one set \( V_i \) we identify the smallest \( V_i \) containing \( m \). The intermediate results from peeling individuals 1 through \( i \) are stored in anterior cutsets, including \( C_{i}^A(g_{V_i}) \), and do not have to be recomputed. In this paper we have introduced a second type of cutset, called a posterior cutset, together with an algorithm for its efficient computation. The posterior cutset \( C_{i}^P(g_{V_i}) \) contains the intermediate results from peeling all individuals that did not contribute to \( C_{i}^A(g_{V_i}) \) and are not contained in the set \( V_i \). Thus, by making use of the intermediate results stored in both \( C_{i}^A(g_{V_i}) \) and \( C_{i}^P(g_{V_i}) \), only calculations associated with peeling individuals in \( V_i \) (except \( m \)) need to be repeated to compute the numerator \( L_{g_n=x} \) of 13. For any \( m \) that is not in any \( V_i \) the expression used to compute genotype probabilities (14) cannot be written as a product of a single anterior and posterior. However, any of the anterior the posterior cutsets used in 14 can be computed efficiently. Thus, this new peeling based algorithm provides an efficient method to compute marginal genotype probabilities for an arbitrary member of a pedigree with loops. The computational cost of obtaining posterior genotype probabilities for all members of a pedigree would approximately be equal to twice that of computing the likelihood because computing the likelihood only requires computing the anterior cutsets while computing all genotype probabilities would require computing the posterior cutsets also. As stated by Jensen and Kong [15], a peeling based algorithm would be more accessible to researchers in genetics than the currently available junction-tree methods [10].

Throughout this paper the likelihood was written as a sum over genotype variables. However, when the genotype of an individual is defined over \( k \) loci, the number of genotypes increases exponentially with \( k \). In such situations, writing the likelihood as a sum over allele state and origin
allele variables may lead to more efficient computations [12]. Algorithms presented in this paper can be used to calculate the posterior allele state and allele origin probabilities by peeling over allele state and allele origin variables.

### Competing interests
The authors declare that they have no competing interests.

### Authors’ contributions
LRT and RLF developed and programmed the algorithm in C++. The analysis of the real cattle pedigree was performed by LRT. KJA contributed to the C++ implementation of the algorithm. The manuscript was prepared by LRT and RLF. All authors have read and approved the final manuscript.

### Appendix
The pedigree given in Figure 2 will be used to illustrate the theoretical concepts discussed above.

First we show how to use the Elston-Stewart algorithm to compute the likelihood for a genetic model given this pedigree. Next we describe how to calculate marginal posterior genotype probabilities for an arbitrary member of this pedigree using the efficient algorithm described above.

#### Likelihood computations by peeling
As shown in 2, the likelihood given the observed data can be written as

\[
L = \sum_{s_7} \sum_{s_6} \cdots \sum_{s_1} f_7(s_7)f_6(s_6) \times f_5(s_7,s_6,s_5)f_4(s_7,s_6,s_4) \times f_3(s_5,s_4,s_3)f_2(s_5,s_4,s_2)f_1(s_5,s_4,s_1).
\]

In the pedigree given in Figure 2, individuals are numbered according to a suitable peeling sequence. Note that in 25 \(f_1(s_5,s_4,s_1)\) is the only function that involves \(s_1\). Peeling \(s_1\) amounts to computing the sum over \(s_1\) of \(f_1(s_5,s_4,s_1)\), for each combination of the genotypes for individuals 5 and 4, and storing the results of these summations in the anterior cutset

\[
C^A(s_5,s_4) = \sum_{s_1} f_1(s_5,s_4,s_1).
\]

Note that \(C^A(s_5,s_4)\) is a two dimensional table of size \(N_5 \times N_4\), where \(N_5\) and \(N_4\) are the number of possible genotypes for individuals 5 and 4. Replacing the sum over \(s_1\) of \(f_1(s_5,s_4,s_1)\) in 25 with \(C^A(s_5,s_4)\) gives the likelihood expression \(LE_1:\)

\[
L = \sum_{s_7} \sum_{s_6} \cdots \sum_{s_1} f_7(s_7)f_6(s_6) \times f_5(s_7,s_6,s_5)f_4(s_7,s_6,s_4) \times f_3(s_5,s_4,s_3)f_2(s_5,s_4,s_2)C^A(s_5,s_4).
\]

Replacing the sum over \(s_2\) of \(f_2(s_5,s_4,s_2)\) in \(LE_1\) with \(C^A(s_5,s_4)\) gives the likelihood expression \(LE_2:\)

\[
L = \sum_{s_7} \sum_{s_6} \cdots \sum_{s_1} f_7(s_7)f_6(s_6) \times f_5(s_7,s_6,s_5)f_4(s_7,s_6,s_4) \times f_3(s_5,s_4,s_3)C^A(s_5,s_4)C^A(s_5,s_4).
\]

### Table 2: Marker allele scores for two markers flanking the causative recessive locus.

| Individual | M1A1 | M1A2 | M2A1 | M2A2 |
|-----------|------|------|------|------|
| A1        | 1    | 1    | 3    | 1    |
| A2        | 2    | 2    | 2    | 2    |
| A3        | 3    | 3    | 2    | 2    |
| A4        | 2    | 1    | 1    | 2    |
| A5        | 3    | 1    | 2    | 1    |
| A6        | 3    | 1    | 2    | 1    |
| A7        | 2    | 1    | 2    | 1    |
| A8        | 2    | 3    | 2    | 2    |
| A9        | 2    | 2    | 2    | 2    |
| A10       | 2    | 2    | 2    | 2    |
| A11       | 0    | 0    | 0    | 0    |
| A12       | 2    | 1    | 2    | 1    |
| A13       | 0    | 0    | 0    | 0    |
| A14       | 0    | 0    | 0    | 0    |
| A15       | 2    | 1    | 2    | 1    |
| A16       | 2    | 1    | 2    | 1    |
| A17       | 2    | 3    | 2    | 2    |
| A18       | 2    | 3    | 2    | 2    |
| A19       | 2    | 1    | 2    | 1    |
| A20       | 0    | 0    | 2    | 1    |
| A21       | 2    | 2    | 2    | 2    |
| A22       | 2    | 2    | 2    | 2    |
| A23       | 2    | 2    | 2    | 2    |
| A24       | 2    | 2    | 2    | 2    |
| A25       | 2    | 2    | 2    | 2    |
| A26       | 2    | 3    | 2    | 2    |

Each marker has three alleles coded as 1, 2 and 3, with 0 denoting a missing value.
Note that in \( LE_2 \) \( f_3(g_5, g_4, g_3) \) is the only function that involves \( g_3 \). Therefore, the anterior cutset for 3 (obtained by peeling \( g_3 \)) is

\[
C_3^A(g_5, g_4) = \sum_{g_3} f_3(g_5, g_4, g_3).
\] (26)

Replacing the sum over \( g_3 \) of \( f_3(g_5, g_4, g_3) \) in \( LE_2 \) with \( C_3^A(g_5, g_4) \) gives the likelihood expression \( LE_3 \):

\[
C_3^A(g_5, g_4) = \sum_{g_3} f_3(g_5, g_4, g_3).
\]

Note that in \( LE_3 \) not only \( f_4(g_7, g_6, g_5) \), but also \( C_3^A(g_5, g_4), C_2^A(g_5, g_4) \) and \( C_1^A(g_5, g_4) \) involve \( g_4 \). Thus, peeling \( g_4 \) yields the following anterior cutset

\[
C_4^A(g_7, g_6, g_5) = \sum_{g_4} f_4(g_7, g_6, g_4)C_3^A(g_5, g_4)C_2^A(g_5, g_4)C_1^A(g_5, g_4).
\] (27)

The resulting anterior cutset \( C_4^A(g_7, g_6, g_5) \) is a three dimensional table of size \( N_7 \times N_6 \times N_5 \), where \( N_7, N_6 \) and \( N_5 \) are the number of possible genotypes for individuals 7, 6 and 5. \( C_4^A(g_7, g_6, g_5) \) replaces in \( LE_4 \) the factors \( f_4(g_7, g_6, g_4), C_3^A(g_5, g_4), C_2^A(g_5, g_4) \) and \( C_1^A(g_5, g_4) \) summed over \( g_4 \). Thus, the likelihood expression \( LE_4 \) becomes

\[
L = \sum_{g_7} \sum_{g_6} \sum_{g_5} f_7(g_7)f_6(g_6)f_5(g_5, g_6, g_5)C_4^A(g_7, g_6, g_5).
\]
Note that in $LE_4$ both $f_5(g_7, g_6, g_5)$ and $C_4^A(g_7, g_6, g_5)$ involve $g_5$. Peeling $g_5$ yields the following anterior cutset

$$C_5^A(g_7, g_6) = \sum_{g_5} f_5(g_7, g_6, g_5) C_4^A(g_7, g_6, g_5).$$

(28)

This cutset replaces in $LE_4$ the factors $f_5(g_7, g_6, g_5)$ and $C_4^A(g_7, g_6, g_5)$ summed over $g_5$. Thus, the likelihood expression $LE_5$ becomes

$$L = \sum_{g_5} \sum_{g_6} f_7(g_7) f_6(g_6) C_5^A(g_7, g_6).$$

In $LE_5$ both $f_6(g_6)$ and $C_5^A(g_7, g_6)$ involve $g_6$. Peeling $g_6$ yields the following anterior cutset

$$C_6^A(g_7) = \sum_{g_6} f_6(g_6) C_5^A(g_7, g_6).$$

(29)

By replacing $f_6(g_6)$ and $C_5^A(g_7, g_6)$ summed over $g_6$ with $C_6^A(g_7)$ in $LE_5$, the likelihood expression $LE_6$ becomes

$$L = \sum_{g_6} f_7(g_7) C_6^A(g_7).$$

Note, however, that the anterior cutset obtained by peeling $g_6$ yields the numerical value

$$C_7^A() = \sum_{g_6} f_7(g_7) C_6^A(g_7).$$

(30)

and thus the likelihood expression $LE_7$:

$$L = C_7^A().$$

**Genotype probability computations**

Recall that for an arbitrary member of the pedigree (e.g. individual 3) we can calculate marginal genotype probabilities as follows

$$Pr(g_3 = x) = \frac{L_{g_3=x}}{L},$$

(31)

where $L_{g_3=x}$ is the likelihood computed with $g_3$ fixed at $x$. As discussed earlier, using this procedure to compute marginal genotype probabilities for $N$ unknown genotypes of individual 3 requires recomputing the likelihood for the entire pedigree $N$ times. However by writing the likelihood as in 12, these computations can be done efficiently.

Consider computing marginal posterior genotype probabilities for individual 3. Recall that, as shown in 26, $C_3^P(g_5, g_4) = \sum_{g_5} f_3(g_5, g_4, g_3)$. Using this in 12 we obtain

$$L = \sum_{g_5} \sum_{g_4} \sum_{g_3} f_3(g_5, g_4, g_3) C_3^P(g_5, g_4).$$

(32)

Note that 32 can be used to calculate the denominator of 31, while the numerator of 31 can be obtained by fixing $g_3$ in 32 at $x$. To complete the calculations, however, we need to compute $C_3^P(g_5, g_4)$. This is done using the recursive procedure described previously as shown below.

Step 1 of the procedure is to compute anterior cutsets for all individuals in the pedigree, and this has already been done. Following step 2, we determine that $C_4^A(g_5, g_4)$ contributes to the computation of $C_4^A(g_7, g_6, g_5)$ (see equation 27). Following step 3, $C_4^A(g_7, g_6, g_5) = \sum_{g_7} f_4(g_7, g_6, g_5) C_4^A(g_7, g_6, g_5)$ is replaced with $C_3^P(g_5, g_4)$ in 27 and, for each value of $g_4$ and $g_5$, the sum over $g_7$ and $g_6$ is computed to obtain

$$C_3^P(g_5, g_4) = \sum_{g_7} \sum_{g_6} f_4(g_7, g_6, g_5) C_4^A(g_7, g_6, g_5).$$

(33)

Following step 4, note that $C_5^A(g_7, g_6, g_5)$ is not computed yet. Thus, steps 2, 3 and 4 are repeated as follows.

Following step 2, we determine that $C_4^A(g_7, g_6, g_5)$ contributes to the computation of $C_5^A(g_7, g_6)$ (see equation 28). Following step 3, $C_5^A(g_7, g_6, g_5) = \sum_{g_6} f_5(g_7, g_6, g_5) C_5^A(g_7, g_6)$ is replaced with $C_3^P(g_5, g_4)$ in 28 and, for each value of $g_4$ and $g_5$, the sum over $g_7$ and $g_6$ is computed to obtain

$$C_3^P(g_5, g_4) = \sum_{g_6} \sum_{g_7} f_5(g_7, g_6, g_5) C_5^A(g_7, g_6).$$

(34)

Following step 4, note that $C_5^P(g_7, g_6)$ is not computed yet. Thus, steps 2, 3 and 4 are repeated as follows.

Following step 2, we determine that $C_4^A(g_7, g_6, g_5)$ contributes to the computation of $C_5^A(g_7, g_6)$ (see equation 29).
Following step 3, $C_6^A(g_7, g_6)$ is replaced with $C_6^P(g_7)$ in 29 and, for each value of $g_7$ and $g_6$ we obtain

$$C_5^P(g_7, g_6) = f_6(g_6)C_6^P(g_7).$$

Following step 4, note that $c_6^P(g_7)$ is not computed yet. Thus, steps 2, 3 and 4 are repeated as follows.

Following step 2, we determine that $C_7^A(g_7)$ contributes to the computation of $C_7^A()$ (see equation 30).

Following step 3, $C_6^A(g_7)$ is replaced with $C_7^P()$ in 30 and, for each value of $g_7$ we obtain

$$C_6^P(g_7) = f_7(g_7)C_7^P().$$

Following step 4, note that $C_7^P() = 1.0$, and thus the calculations for $C_6^P(g_7)$ can be completed. Now using $C_6^P(g_7)$, the calculations for $C_5^P(g_7, g_6)$ can be completed, and using $C_5^P(g_7, g_6)$, the calculations for $C_4^P(g_7, g_6, g_5)$ can be completed. Finally, using $C_4^P(g_7, g_6, g_5)$, the calculations for $C_3^P(g_5, g_4)$ can be completed.

**Additional material**

**Additional file 1**

A numerical example to illustrate algorithm to detect loops in a pedigree. Algorithm to detect loops in a pedigree.

Click here for file: [http://www.biomedcentral.com/content/supplementary/1297-9686-41-52-S1.pdf](http://www.biomedcentral.com/content/supplementary/1297-9686-41-52-S1.pdf)

**Acknowledgements**

The authors would like to thank James Reecy and James Koltes for providing the marker and phenotypic data for the real cattle pedigree discussed in this article. RLF is supported by the United States Department of Agriculture, National Research Initiative grant USDA-NRI-2007-35205-17862.

**References**

1. Elston RC, Stewart J: A general model for the genetic analysis of pedigree data. *Hum Hered* 1971, 21:523-542.
2. Lange K, Elston RC: Extension to pedigree analysis. I. Likelihood calculations for simple and complex pedigrees. *Hum Hered* 1975, 25:95-105.
3. Cannings C, Thompson EA, Skolnick MH: Probability functions on complex pedigrees. *Adv Appl Prob* 1978, 10:26-61.
4. Thomas A: Approximate computation of probability functions for pedigree analysis. *IMA J Math Appl Med Biol* 1986, 3:157-166.
5. Lander ES, Green P: Construction of multilocus genetic linkage maps in humans. *Proc Natl Acad Sci USA* 1987, 84(8):2363-2367.
6. Heath S: Markov chain Monte Carlo segregation and linkage analysis for oligogene models. *Am J Hum Genet* 1997, 61:748-760.
7. Fernández SA, Fernando RL, Guhlbrandt B, Tozír LR, Carriquiry AL: Sampling genotypes in large pedigrees with loops. *Genet Sel Evol* 2001, 33:337-367.
8. Fernando R, Tozír L, Pita F, Stricker C, Abraham K: Algorithms to compute allele state and origin probabilities for QTL mapping. *8th World Congress Genet Appl Livest Prod* 2006.
9. Fernando RL, Stricker C, Elston RC: An efficient algorithm to compute the posterior genotypic distribution for every member of a pedigree without loops. *Theor Appl Genet* 1993, 87:89-93.
10. Lauritzen SL, Sheehan NA: Graphical models for genetic analysis. Statist Sci 2003, 18:489-514.
11. Thompson E: Pedigree Analysis in Human Genetics The Johns Hopkins University Press, Baltimore; 1986.
12. Fishelson M, Geiger D: Exact genetic linkage computations for general pedigrees. *Bioinformatics* 2002, 18:S189-S198.
13. Cannings C, Thompson EA, Skolnick MH: The recursive derivation of likelihoods on complex pedigrees. *Adv Appl Prob* 1976, 8:622-625.
14. Lange K, Boehnke M: Extensions to pedigree analysis. V. Optimal calculation of mendelian likelihoods. *Hum Hered* 1983, 33:291-301.
15. Jensen CS, Kong A: Blocking Gibbs sampling for linkage analysis in large pedigrees with many loops. *Am J Hum Genet* 1999, 65:885-901.