Efficient computation of genotype probabilities for loci with many alleles: I. Allelic peeling

R. M. Thallman[1], G. L. Bennett, J. W. Keele, and S. M. Kappes

USDA, ARS, Roman L. Hruska U.S. Meat Animal Research Center, Clay Center, NE 68933-0166

ABSTRACT: Genetic marker data are likely to be obtained from a relatively small proportion of the individuals in many livestock populations. Information from genetic markers can be extrapolated to related individuals without marker data by computing genotype probabilities using an algorithm referred to as peeling. However, genetic markers may have many alleles and the number of computations in traditional peeling algorithms is proportional to the number of alleles raised to the sixth or eighth power, depending on pedigree structure. An alternative algorithm for computing genotype probabilities of marker loci with many alleles in large, nonlooped pedigrees with incomplete marker data is presented. The algorithm is based on recursive computations depending on alleles instead of genotypes, as in traditional peeling algorithms. The number of computations in the allelic peeling algorithm presented here is proportional to the square of the number of alleles, which makes this algorithm more computationally efficient than traditional peeling for loci with many alleles. Memory requirements are roughly proportional to the number of individuals in the pedigree and the number of alleles. The recursive allelic peeling algorithm cannot be applied to pedigrees that include full sibs or loops. However, it is a preliminary step toward a more complex and encompassing iterative approach to be described in a companion paper.

Key Words: Genetic Analysis, Genetic Markers, Pedigree, Statistical Genetics

J. Anim. Sci. 2001. 79:26–33

Introduction

Genetic marker information in livestock populations is expected to increase rapidly. Marker information used for identifying QTL and linkage mapping has primarily been collected from large experimental and industry families with three-generation pedigrees and nearly complete marker data (Rohrer and Keele, 1998; Zhang et al., 1998; Stone et al., 1999). Much more information could be gleaned by calculating genotypic probabilities for individuals with missing marker data and tracking markers over an extended pedigree in commercial or long-term experimental populations. This would effectively tie some large families together as well as including many individuals in smaller families.

The method of "peeling" for the calculation of genotype probabilities is based on ideas formulated by Elston and Stewart (1971) and has been extended (Lange and Elston, 1975; Cannings et al., 1978; Fernando et al., 1993). Peeling has been used extensively in human genetics, but primarily on pedigrees with fewer than 100 individuals. Applications of peeling in livestock pedigrees (van Arendonk et al., 1989; Kerr and Kinghorn, 1996; Wang et al., 1996) have focused on models with two alleles and three genotypes. Monte Carlo methods of pedigree analysis (Guo and Thompson, 1992; Uimari et al., 1996) and peeling are both computationally demanding for large, complex pedigrees with many marker alleles.

The objective of this research was to reformulate the method of peeling to make it more computationally efficient for a single marker locus with many alleles (e.g., microsatellites) in a large population. This method has been extended (Thallman et al., 2001) to handle several practical situations that occur in the analysis of genetic markers: looped pedigrees, errors in marker data, and computation of probabilities that summarize the segregation pattern.

Materials and Methods

In the analysis of marker data in simple pedigrees, the ordered genotypes (dam allele, sire allele) of individuals are often "inferred" based on the rules of Mendelian inheritance. In some situations, there is not enough information to infer the ordered genotype with certainty; these genotypes may be considered uninformative. In the context of peeling, the genotypes are always considered unobservable and therefore unknown, but inferences

[1] Correspondence. P.O. Box 166 (phone: 402-762-4261; fax: 402-762-4173; E-mail: thallman@email.marc.usda.gov).

Received March 9, 2000.

Accepted August 10, 2000.

about the genotypes are obtained in the form of probabilities. If a locus has A alleles, then there are A^2 possible ordered genotypes and the probability of each genotype, conditional on marker data, can be computed for an individual of interest. If the ordered genotype can be inferred based on the rules of Mendelian inheritance, then that genotype will have a probability of one and the remaining genotypes will each have probability of zero. In cases with less information, many or all of the genotypes may have nonzero probabilities. Peeling allows inferences to be readily obtained from marker data that are many generations removed from an individual.

Probabilities of possible genotypes of an individual with no marker data are theoretically dependent on all marker data in the pedigree. Expressions for computing these quantities directly from the marker data are not computationally efficient and can be difficult or impossible to generalize in complex pedigrees. The method of peeling solves this problem by breaking the calculations into a series of simpler calculations that are applied first to founders and nonparents and subsequently to individuals with both parents and progeny. Peeling algorithms use recursive formulas that involve only the individual's parents and progeny. Consequently, the recursive formulas are very general, even in large pedigrees.

We refer to the traditional peeling algorithms (Lange and Elston, 1975; van Arendonk et al., 1989; Fernando et al., 1993) as "genotypic peeling," because they are based on recursive relationships among probabilities of genotypes of individuals. We propose an alternative algorithm, "allelic peeling," based on probabilities of alleles transmitted from parent to offspring. Specifying the recursive relationships in terms of alleles rather than genotypes greatly improves the computational efficiency for loci with many alleles, such as microsatellite marker loci.

Definitions and Notation

We assume a pedigreed population with all genetic relationships among individuals known and all common ancestors included in the pedigree. Figure 1 shows a simple pedigree that will be useful in describing the algorithm. Each individual in the pedigree has an (unobservable) ordered genotype, $g_i = [a_{id}, a_{is}]$, where a_{id} is the allele individual i inherited from its dam, d, and a_{is} is the allele i inherited from its sire, s. Some of the individuals in the pedigree have marker data, considered to be phenotypes (Lincoln and Lander, 1992). The relationship between phenotypes and genotypes is specified by the penetrance function (or genetic model).

An individual, x, is connected to i if there is a path that starts at i and ends at x, regardless of the direction of the arrows. If x is connected to i, g_i may be statistically dependent on g_x. The pedigree can be divided into "parental" and "progeny" subsets relative to i. The parental subset relative to i through its parent, d, includes all individuals that are connected to i by at least one path that includes parent d. The progeny subset relative to i through its progeny, k, includes all individuals that are connected to i by at least one path that includes progeny k. The term "connected" is more inclusive than genetic relationship. For example, in Figure 1, b is included in the parental subset relative to i through d, although b is not related to i. The algorithm does not require that the various subsets of individuals be listed explicitly, because the peeling algorithm uses these subsets automatically.

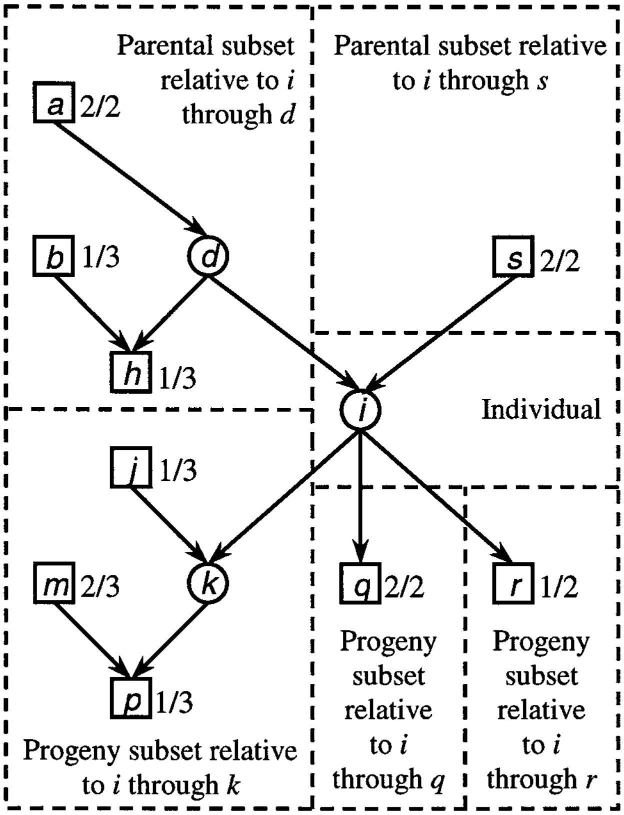
Parental subset relative to i through d
Parental subset relative to i through s
a 2/2
b 1/3
d
s 2/2
h 1/3
i
Individual
j 1/3
m 2/3
k
q 2/2
r 1/2
p 1/3
Progeny subset relative to i through k
Progeny subset relative to i through q
Progeny subset relative to i through r

Figure 1. Partitioning a pedigree without loops into parental, individual, and progeny subsets relative to individual *i*. The marker phenotype is represented by the pair of numbers to the right of the individual. For example, the phenotype of individual *a* is 2/2.

For convenience, we assume that all individuals in the pedigree are connected to one another. If a population does contain unconnected subsets, then each subset can be analyzed as an independent pedigree.

In Figure 1, all the subsets are independent other than through their connection with i (independent subsets conditional on i). There are many common situations in livestock pedigrees that may cause the subsets to be dependent. Inbreeding and mating a sire to genetically related dams are examples. These situations create "loops" in the pedigree (Cannings et al., 1978). A loop occurs when an individual can be connected to itself, through two different parents and(or) progeny. For example, Figure 1 does not contain any loops, but if it was modified so that b was the sire of m, then i would be

Table 1. Notation for probability distributions and likelihoods used in allelic peeling

Symbol[a]	Name	Property of	Elements sum to one?	Dimensions[b]	Row index[c]	Column index[c]	Description
$\boldsymbol{\pi}$	Allele frequencies	Population	Yes	A × 1	a		Prior probability of allele a
$\mathbf{M}(i)$	Penetrance matrix	Individual	No	A × A	a_{id}	a_{is}	Likelihood of the phenotype of i conditional of $\mathrm{g}_i = [\mathrm{a}_{id}, \mathrm{a}_{is}]$
$\mathbf{P}(ki)$	Parental prior distribution	Meiosis	Yes	A × 1	a_{ki}		Probability of allele a_{ki} having been transmitted from i to k conditional on data in the parental subset relative to k through i
$\mathbf{L}(ki)$	Progeny likelihood	Meiosis	No	A × 1	a_{ki}		Scaled likelihood of data in the progeny subset relative to i through k conditional on allele a_{ki} having been transmitted from i to k
$\mathbf{G}(i)$	Genotype distribution	Individual	Yes	A × A	a_{id}	a_{is}	Probability that $\mathrm{g}_i = [\mathrm{a}_{id}, \mathrm{a}_{is}]$ conditional on all marker data in the pedigree

[a] The parentheses enclose the identification of the individual or meiosis to which the matrix pertains. The values i and ki are placeholders, not specific individuals or meioses.

[b] Rows × columns. A is the number of alleles.

[c] The alleles i inherited from its dam and sire are represented by a_{id} and a_{is}, respectively. The allele transmitted through meiosis ki is represented by a_{ki}.

included in a loop because the path *i-k-p-m-b-h-d-i* would then connect i to itself through k and d. In this article, we assume that there are no loops in the pedigree. Thallman et al. (2001) explain how to apply allelic peeling to pedigrees that contain loops.

In genetic linkage analysis, the basic unit of information is the meiosis, rather than the individual. In the linkage analysis literature, it is common to refer to informative meioses, recombinant meioses, and nonrecombinant meioses (Ott, 1999). We use the term *meiosis* in a similar manner but define it more precisely as a parent-offspring pair. Meioses connect pairs of individuals and correspond to the arrows in the pedigree in Figure 1. Meioses in the pedigree are identified by the pair of italicized, lowercase letters corresponding to the offspring and the parent in the meiosis (e.g., ki refers to the meiosis from parent i to offspring k).

One objective of peeling is to compute the probability that the individual of interest has each of the A^2 possible ordered genotypes, conditional on marker data. The set of these A^2 genotype probabilities is the genotype distribution. We represent the genotype distribution as an A × A matrix with rows corresponding to the allele inherited from the dam and columns corresponding to the allele inherited from the sire. We present the peeling formulas in matrix form so that they pertain to entire probability distributions rather than to only the probabilities of individual alleles or genotypes.

Definition of Recursive Elements

Table 1 contains the symbols and definitions of the main probabilities used in allelic peeling. The penetrance function, $\mathbf{M}(i)$, is used to relate the genotype to the phenotype according to the genetic model for the locus. Specifically, it is an A × A matrix of likelihoods of the phenotype of i conditional on each possible genotype of i. The information contained in the phenotype of an individual is summarized by and enters the peeling algorithm through the penetrance matrix of the individual. In the analysis of marker loci, a complete penetrance model is typically used (i.e., the phenotype is completely determined by the genotype). The complete penetrance model for autosomal loci is assumed for this paper. For example, in Figure 1 with three alleles at the locus, the phenotype of r is 1/2, so elements 1, 2 and 2, 1 of $\mathbf{M}(r)$ are equal to one and all other elements of the 3×3 matrix are equal to zero. Because d does not have a phenotype, $\mathbf{M}(d)$ is a matrix filled with ones.

The peeling algorithm is made recursive by functions of subsets of the marker data. One of these is the parental prior distribution for a meiosis, $\mathbf{P}(ki)$, which is a column vector of length A and is described in more detail subsequently as well as in Table 1.

The other main function involved in the recursion is the scaled progeny likelihood for meiosis id, $\mathbf{L}(id)$, which is a column vector of length A proportional to the likelihood of marker phenotypes connected to d through its progeny, i, conditional on each possible allele at the locus having been transmitted from d to i. Because $\mathbf{L}(id)$ is a vector of scaled likelihoods, its elements do not sum to one. The scaled progeny likelihood for the meiosis of maternal origin is calculated as

$$\mathbf{L}(id) = \mathrm{c_L}(id)^{-1} \left\{\mathbf{M}(i) \circ \prod_{t \epsilon \mathrm{progeny}(i)}^{\circ} [0.5 \cdot \mathbf{L}(ti) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ti)'] \right\} \cdot \mathbf{P}(is) \qquad [1]$$

where

$$\mathrm{c_L}(id) = \boldsymbol{\pi}' \cdot \left\{\mathbf{M}(i) \circ \prod^{\circ}_{t \in \text{progeny}(i)} [0.5 \cdot \mathbf{L}(ti) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ti)']\right\} \cdot \mathbf{P}(is)$$

The operator · represents standard matrix or scalar multiplication, whereas the operator ∘ represents elementwise multiplication of matrices. The multiple product is elementwise over each of the progeny of i and is eliminated from the formula if i has no progeny. The operator ′ indicates matrix transposition. The constant, **1**, is a column vector of length A filled with ones. The scalar, $\mathrm{c_L}(id)$, is a scaling factor used to prevent numeric underflows in large pedigrees. In [1], and all of the equations that follow, i, d, s, and k do not refer to specific individuals in Figure 1, but instead refer to any individual and its dam, sire, and progeny, respectively.

In [1], $\mathbf{L}(id)$ summarizes the information about a_{id} contained in the progeny subset relative to the dam, d, through its progeny, i. This is accomplished by converting the scaled likelihood of phenotypes connected to i through t conditional on a_{ti} into the scaled likelihood of the same phenotypes conditional on a_{id} and a_{is}, which is $[0.5 \cdot \mathbf{L}(ti) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ti)']$. The progeny subsets of phenotypes relative to i through different progeny and the phenotype of i are mutually independent conditional on both a_{id} and a_{is}. Therefore, their scaled likelihoods conditional on a_{id} and a_{is} can simply be multiplied together. The product is reduced to $\mathbf{L}(id)$, the scaled likelihood of the same phenotypes conditional only on a_{id} by the matrix multiplication by $\mathbf{P}(is)$, which is equivalent to summation over the possible values of a_{is}. The union of the subsets of phenotypes considered in the right-hand side of [1] is the progeny subset of d through i.

The scaled progeny likelihood for the meiosis of paternal origin is calculated as

$$\mathbf{L}(is) = \mathrm{c_L}(is)^{-1}\left\{\mathbf{M}(i) \circ \prod^{\circ}_{t \in \text{progeny}(i)} [0.5 \cdot \mathbf{L}(ti) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ti)']\right\}' \cdot \mathbf{P}(id) \qquad [2]$$

where

$$\mathrm{c_L}(is) = \boldsymbol{\pi}' \cdot \left\{\mathbf{M}(i) \circ \prod^{\circ}_{t \in \text{progeny}(i)} [0.5 \cdot \mathbf{L}(ti) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ti)']\right\}' \cdot \mathbf{P}(id)$$

The only difference between [1] and [2] is that the A × A matrix in braces is transposed when computing the result for the paternal meiosis. For the models described in this paper for autosomal loci, this matrix is symmetric, so [1] and [2] are equal. However, for several extensions to the method, including sex-linked loci and peeling conditional on linked loci, the matrix is asymmetric. The matrix transposition in [2] ensures that it is the maternally inherited allele that is marginalized out of the likelihood when computing $\mathbf{L}(is)$.

The terms in [1] and [2] that are equal to 0.5 are prior probabilities of the associated meiosis having inherited either the parent's maternally or paternally derived allele. When peeling conditional on linked loci, they can take values different from 0.5, and therefore they are included in the formulas.

The parental prior distribution for a meiosis, $\mathbf{P}(ki)$, is a column vector of length A, with elements containing the probabilities of each allele at the locus having been transmitted through meiosis ki conditional on marker phenotypes connected to k through its parent, i. It is computed recursively as

$$\begin{aligned}\mathbf{P}(ki) = \mathrm{c_P}(ki)^{-1}\Big[&0.5 \cdot \mathbf{P}(id) \circ \Big(\{\mathbf{M}(i) \circ \prod^{\circ}_{\substack{t \in \text{progeny}(i)\\ t \neq k}} [0.5 \cdot \mathbf{L}(ti) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ti)']\} \cdot \mathbf{P}(is)\Big)\\ &+ 0.5 \cdot \mathbf{P}(is) \circ \Big(\{\mathbf{M}(i) \circ \prod^{\circ}_{\substack{t \in \text{progeny}(i)\\ t \neq k}} [0.5 \cdot \mathbf{L}(ti) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ti)']\}' \cdot \mathbf{P}(id)\Big)\Big]\end{aligned} \qquad [3]$$

where

$$\begin{aligned}\mathrm{c_P}(ki) = \sum\Big[&0.5 \cdot \mathbf{P}(id) \circ \Big(\{\mathbf{M}(i) \circ \prod^{\circ}_{\substack{t \in \text{progeny}(i)\\ t \neq k}} [0.5 \cdot \mathbf{L}(ti) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ti)']\} \cdot \mathbf{P}(is)\Big)\\ &+ 0.5 \cdot \mathbf{P}(is) \circ \Big(\{\mathbf{M}(i) \circ \prod^{\circ}_{\substack{t \in \text{progeny}(i)\\ t \neq k}} [0.5 \cdot \mathbf{L}(ti) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ti)']\}' \cdot \mathbf{P}(id)\Big)\Big]\end{aligned}$$

The terms, $\mathbf{P}(id)$ and $\mathbf{P}(is)$ are the parental prior distributions of the alleles transmitted to i by i's dam, d, and sire, s. The elements of $\mathbf{P}(ki)$ are forced to sum to one by the scaling factor, $\mathrm{c_P}(ki)$. The summation in the expression for $\mathrm{c_P}(ki)$ is over the A elements of the vector.

The parental prior distribution, **P**(ki), summarizes the information about a_{ki} contained in the parental subset relative to k through i. In [3], **P**(id) is multiplied elementwise by a term that is the same as **L**(id) except that it excludes the progeny subset relative to i through k. The subsets of phenotypes considered by **P**(id) and the modified form of **L**(id) are disjoint (independent conditional on a_{id}) and their union is the parental subset of phenotypes relative to k through i. Therefore, after scaling by $c_P(ki)$, this product is equal to the parental prior distribution of a_{ki} conditional on k having inherited the allele that i inherited from d (a_{ki} being identical by descent to a_{id}). The 0.5 that this product is multiplied by is the prior probability of that condition having been met. The remainder of [3] contains the parental prior distribution conditional on k having inherited the allele from s and the probability of that condition having been met. The progeny subset relative to i through k is excluded from the multiple product because it is not part of the parental subset relative to k through i, although the other progeny subsets relative to i are. For example, in Figure 1, the parental subset relative to k through i is the union of {a, b, d, h}, {s}, and {i, q, r}, which are summarized by **P**(id), **P**(is), and the remaining term in [3], respectively. Element 2 of **P**(ki) contains the probability that allele 2 was transmitted from i to k conditional on the phenotypes of a, b, h, s, q, and r (i and d are also included in the parental subset relative to k through i but do not have phenotypes).

If i is a founder, then its dam and sire are not included in the pedigree, so **P**(id) and **P**(is) are not defined, but are replaced in [3] with $\boldsymbol{\pi}$, which is the vector of population allele frequencies, a parameter of the analysis. If only one of i's parents is included in the pedigree, then the parental prior distribution for that parent is used and $\boldsymbol{\pi}$ is used for the meiosis from the unknown parent.

Genotype Probabilities

The genotype distribution of i, **G**(i), is an A × A matrix with elements that sum to one. It summarizes what can be inferred about the genotype of i conditional on all the marker phenotypes in the pedigree. The rows and columns are indexed by a_{id} and a_{is}, respectively, and each element contains the probability that i has the corresponding genotype. For example, row 1, column 2 of **G**(i) contains the joint probability that i inherited allele 1 from its dam and allele 2 from its sire. The expression for computing **G**(i) is

$$\mathbf{G}(i) = c_G(i)^{-1}[\mathbf{P}(id) \cdot \mathbf{P}(is)'] \circ \left\{\mathbf{M}(i) \circ \prod_{t \in \text{progeny}(i)}^{\circ} [0.5 \cdot \mathbf{L}(ti) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ti)']\right\} \qquad [4]$$

where

$$c_G(i) = \sum\left([\mathbf{P}(id) \cdot \mathbf{P}(is)'] \circ \left\{\mathbf{M}(i) \circ \prod_{t \in \text{progeny}(i)}^{\circ} [0.5 \cdot \mathbf{L}(ti) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ti)']\right\}\right)$$

and the summation in $c_G(i)$ is over the elements of the matrix.

In [4], **G**(i) summarizes the information about the genotype of i contained in all the data. It is computed from the information about the two alleles in i's genotype contained in the parental subsets relative to i and the information about the genotype of i contained in the phenotypes of i and in the progeny subsets relative to i. The union of the subsets of phenotypes considered in the right-hand side of [4] is the set of all phenotypes in the pedigree.

Recursive Algorithm

Wang et al. (1996) gave a detailed explanation of genotypic terminal peeling, which is a noniterative algorithm for peeling pedigrees that do not contain loops. Terminal parents have one progeny and no parents in the pedigree. Terminal progeny have one parent and no progeny. The terminal individuals are "peeled" away (temporarily removed) from the pedigree with the information that they contain transferred to the core of the pedigree. Peeling a layer of terminal individuals results in a new layer of individuals becoming terminal, and the process is continued until all of the information is concentrated on a single individual, at which point the genotype distribution of that individual can be computed.

To peel the example pedigree in Figure 1 with allelic peeling, the first step is to transfer the information on the terminal parents, a, b, s, j, and m, to their connecting meioses using [3] to compute **P**(da), **P**(hb), **P**(is), **P**(kj), and **P**(pm), respectively, as shown in Table 2. In this step, the parental prior distributions are replaced with $\boldsymbol{\pi}$ (because the terminal parents are founders) and the multiple products are all null (because each of the terminal individuals is connected to the pedigree by only one progeny). Individuals a, b, s, j, and m are now peeled so that h and p are now considered terminal individuals. Next, the terminal progeny, h, p, q, and r, are peeled using [1] to compute **L**(hd), **L**(pk), **L**(qi), and **L**(ri), respectively. Now the terminal individuals are d and k. To peel d, **P**(id) is computed from **P**(da) and **L**(hd) using [3] and replacing the parental prior distribution to d from her dam with $\boldsymbol{\pi}$. To peel k, **L**(ki) is computed from **P**(kj) and **L**(pk) using [1]. At this point, all individuals except i have been peeled, so **G**(i) can be computed from **P**(id), **P**(is), **L**(ki), **L**(qi), and **L**(ri) as shown in Table 2.

Table 2. Computation of the genotype distribution of individual *i* in Figure 1 by allelic peeling

Term	Eq.	Calculation of kernel[a]	Kernel[b]	Scaling factor[c]	Result[d]
$\mathbf{P}(da)$	[3]	$0.5 \cdot \boldsymbol{\pi} \circ [\mathbf{M}(a) \cdot \boldsymbol{\pi}] + 0.5 \cdot \boldsymbol{\pi} \circ [\mathbf{M}(a)' \cdot \boldsymbol{\pi}]$	$\begin{bmatrix}0.00\\0.11\\0.00\end{bmatrix}$	0.11	$\begin{bmatrix}0.00\\1.00\\0.00\end{bmatrix}$
$\mathbf{P}(hb)$	[3]	$0.5 \cdot \boldsymbol{\pi} \circ [\mathbf{M}(b) \cdot \boldsymbol{\pi}] + 0.5 \cdot \boldsymbol{\pi} \circ [\mathbf{M}(b)' \cdot \boldsymbol{\pi}]$	$\begin{bmatrix}0.11\\0.00\\0.11\end{bmatrix}$	0.22	$\begin{bmatrix}0.50\\0.00\\0.50\end{bmatrix}$
$\mathbf{P}(is)$	[3]	$0.5 \cdot \boldsymbol{\pi} \circ [\mathbf{M}(s) \cdot \boldsymbol{\pi}] + 0.5 \cdot \boldsymbol{\pi} \circ [\mathbf{M}(s)' \cdot \boldsymbol{\pi}]$	$\begin{bmatrix}0.00\\0.11\\0.00\end{bmatrix}$	0.11	$\begin{bmatrix}0.00\\0.11\\0.00\end{bmatrix}$
$\mathbf{P}(kj)$	[3]	$0.5 \cdot \boldsymbol{\pi} \circ [\mathbf{M}(j) \cdot \boldsymbol{\pi}] + 0.5 \cdot \boldsymbol{\pi} \circ [\mathbf{M}(j)' \cdot \boldsymbol{\pi}]$	$\begin{bmatrix}0.11\\0.00\\0.11\end{bmatrix}$	0.22	$\begin{bmatrix}0.50\\0.00\\0.50\end{bmatrix}$
$\mathbf{P}(pm)$	[3]	$0.5 \cdot \boldsymbol{\pi} \circ [\mathbf{M}(m) \cdot \boldsymbol{\pi}] + 0.5 \cdot \boldsymbol{\pi} \circ [\mathbf{M}(m)' \cdot \boldsymbol{\pi}]$	$\begin{bmatrix}0.00\\0.11\\0.11\end{bmatrix}$	0.22	$\begin{bmatrix}0.00\\0.50\\0.50\end{bmatrix}$
$\mathbf{L}(hd)$	[1]	$\mathbf{M}(h) \cdot \mathbf{P}(hb)$	$\begin{bmatrix}0.50\\0.00\\0.50\end{bmatrix}$	0.33	$\begin{bmatrix}1.50\\0.00\\1.50\end{bmatrix}$
$\mathbf{L}(pk)$	[1]	$\mathbf{M}(p) \cdot \mathbf{P}(pm)$	$\begin{bmatrix}0.50\\0.00\\0.00\end{bmatrix}$	0.17	$\begin{bmatrix}3.00\\0.00\\0.00\end{bmatrix}$
$\mathbf{L}(qi)$	[1]	$\mathbf{M}(q) \cdot \boldsymbol{\pi}$	$\begin{bmatrix}0.00\\0.33\\0.00\end{bmatrix}$	0.11	$\begin{bmatrix}0.00\\3.00\\0.00\end{bmatrix}$
$\mathbf{L}(ri)$	[1]	$\mathbf{M}(r) \cdot \boldsymbol{\pi}$	$\begin{bmatrix}0.33\\0.33\\0.00\end{bmatrix}$	0.22	$\begin{bmatrix}1.50\\1.50\\0.00\end{bmatrix}$
$\mathbf{P}(id)$	[3]	$0.5 \cdot \boldsymbol{\pi} \circ \big[[0.5 \cdot \mathbf{L}(hd) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(hd)'] \cdot \mathbf{P}(da)\big]$ $+ 0.5 \cdot \mathbf{P}(da) \circ \big[[0.5 \cdot \mathbf{L}(hd) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(hd)']' \cdot \boldsymbol{\pi}\big]$	$\begin{bmatrix}0.13\\0.25\\0.13\end{bmatrix}$	0.50	$\begin{bmatrix}0.25\\0.50\\0.25\end{bmatrix}$
$\mathbf{L}(ki)$	[1]	$[0.5 \cdot \mathbf{L}(pk) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(pk)'] \cdot \mathbf{P}(kj)$	$\begin{bmatrix}2.25\\0.75\\0.75\end{bmatrix}$	1.25	$\begin{bmatrix}1.80\\0.60\\0.60\end{bmatrix}$
$\mathbf{G}(i)$	[4]	$[\mathbf{P}(id) \cdot \mathbf{P}(is)'] \circ [0.5 \cdot \mathbf{L}(ki) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ki)']$ $\circ [0.5 \cdot \mathbf{L}(qi) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(qi)']$ $\circ [0.5 \cdot \mathbf{L}(ri) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ri)']$	$\begin{bmatrix}0.00 & 0.68 & 0.00\\0.00 & 1.35 & 0.00\\0.00 & 0.17 & 0.00\end{bmatrix}$	2.19	$\begin{bmatrix}0.00 & 0.31 & 0.00\\0.00 & 0.62 & 0.00\\0.00 & 0.08 & 0.00\end{bmatrix}$

[a] The specific expression for the kernel (the formula with the scaling factor omitted) using the general equation indicated in the previous column. Terms that do not apply are omitted. For example, because *h* has no progeny, the multiple product over progeny in [1] is omitted from the expression for **L**(*hd*), and because *k* has no phenotype, the penetrance matrix is omitted from the expression for **L**(*ki*).

[b] The results of the expressions in the previous column, based on $\boldsymbol{\pi} = \begin{bmatrix}0.333\\0.333\\0.333\end{bmatrix}$, $\mathbf{M}(r) = \begin{bmatrix}0 & 1 & 0\\1 & 0 & 0\\0 & 0 & 0\end{bmatrix}$, $\mathbf{M}(b) = \mathbf{M}(h) = \mathbf{M}(j) = \mathbf{M}(p) = \begin{bmatrix}0 & 0 & 1\\0 & 0 & 0\\1 & 0 & 0\end{bmatrix}$, $\mathbf{M}(a) = \mathbf{M}(s) = \mathbf{M}(q) = \begin{bmatrix}0 & 0 & 0\\0 & 1 & 0\\0 & 0 & 0\end{bmatrix}$, and $\mathbf{M}(m) = \begin{bmatrix}0 & 0 & 0\\0 & 0 & 1\\0 & 1 & 0\end{bmatrix}$. Other terms required are computed in previous rows.

[c] Computed by summing the elements of the kernel in the previous column (for parental prior and genotypic distributions) or premultiplying it by the transposed vector of allele frequencies (for progeny likelihoods).

[d] Computed by dividing the kernel by the scaling factor.

If genotype probabilities for all the individuals in the pedigree are desired, the peeling sequence is reversed. All of the required quantities are available to compute $\mathbf{P}(ki), \mathbf{L}(id), \mathbf{P}(ri), \mathbf{P}(qi), \mathbf{P}(pk), \mathbf{P}(hd), \mathbf{L}(pm), \mathbf{L}(kj), \mathbf{L}(is)$, $\mathbf{L}(hb)$, and $\mathbf{L}(da)$ in the sequence listed. At this point, the parental prior distributions and progeny likelihoods are both available for each of the meioses in the pedigree so that the genotype distribution for each member of the pedigree can be readily computed using [4].

Under either genotypic or allelic peeling, if the pedigree contained a loop, the loop would remain after all of the terminal individuals were peeled. At this point, the recursive algorithm above would fail, because there would be no entry point into the loop. Cannings et al. (1978) addressed this problem by peeling on sets of individuals instead of single individuals, but this algorithm would not be computationally feasible for the degree of looping often present in large livestock populations.

Discussion

The term *peeling* originates from the idea of removing terminal members of a pedigree recursively by transferring the genotypic information from them to their parents or progeny and then repeating the process until there is only one remaining member of the pedigree. In allelic peeling, the information is transferred to the meiosis that connects the individual to be peeled with the core of the pedigree. This information is in the form of a parental prior distribution or a progeny likelihood relative to the allele transmitted through this meiosis. The parental prior distribution of a meiosis is computed recursively from the parental prior distributions of the parental meioses, the penetrance function of the parent, and the progeny likelihoods of the meioses to sibs. The progeny likelihood of the same meiosis is computed from the penetrance function of the progeny, the progeny likelihoods of the progeny meioses, and the parental prior distributions of the meioses from the mates. Intermediate computations of order A × A related to the genotypes of individuals are required, but because they are computed from terms of order A × 1 and used to compute terms of order A × 1, fewer computations are required than in genotypic peeling.

The parental prior distributions are computed by recursive application of Bayes' theorem in which the prior distribution of population allele frequencies, $\boldsymbol{\pi}$, is conditioned by progressively larger subsets of the data. The information contained in the parental subsets is summarized in the form of probabilities of alleles because the initial value for recursion of parental information (prior allele frequencies) is in this form.

The information contained in the progeny subsets is summarized in the form of likelihoods of data because the initial value for recursion of progeny information (the penetrance matrix) is in this form. This form also makes it easy to combine information from the subsets of data recursively. The likelihood of the union of conditionally independent subsets of data is simply the product of the likelihoods of the respective subsets.

Table 3. Computations required as a function of the number of alleles for allelic peeling compared to genotypic peeling[a]

Computation	Allelic	Genotypic[b]
Parental prior distribution for individual with no full sibs	$O(A^2)$	$O(A^6)$
Parental prior distribution for individual with full sibs	N/A[c]	$O(A^8)$
Progeny likelihood	$O(A^2)$	$O(A^6)$
Genotype distribution	$O(A^2)$	$O(A^2)$

[a] A = number of alleles. O(x) is the number of computing operations "proportional to x."

[b] The algorithm of Fernando et al. (1993) was used.

[c] Allelic peeling cannot be applied recursively to pedigrees with full sibs. However, in the iterative allelic peeling algorithm described by Thallman et al. (2001), the parental prior distribution is computed in time $O(A^2)$ even when there are full sibs in the pedigree.

The term $\mathbf{P}(is)$ enters into the right-hand side of the expression for $\mathbf{L}(id)$ in [1] because the penetrance function and progeny likelihoods provide information about which two alleles are in the genotype of i, and $\mathbf{P}(is)$ provides information about which of those two alleles i inherited from s and, consequently, which is more likely to have been inherited from d. For example, in Figure 1, the phenotype of p indicates that it has alleles 1 and 3, but $\mathbf{P}(pm)$ indicates that m could contribute only alleles 2 or 3 to i, and therefore $\mathbf{L}(pk)$ indicates that p must have inherited allele 1 from k (Table 2).

In [4], $\mathbf{P}(id) \cdot \mathbf{P}(is)'$ can be viewed as a prior genotype distribution (conditioned by data in the parental subsets relative to i) that is then conditioned by the likelihood of data on i and in the progeny subsets relative to i using Bayes' theorem to obtain $\mathbf{G}(i)$, the posterior genotype distribution conditional on all the phenotypes. The application of Bayes' theorem is facilitated by the fact that the data in parental subsets are summarized in the form of probabilities of alleles and the data in progeny subsets are summarized in the form of likelihoods of data conditional on alleles.

Full Sibs

Full sibs generate loops and therefore the recursive algorithm for allelic peeling cannot be used on pedigrees that include full sibs. The two parental subsets of an individual and its full sib overlap, and therefore it is possible for the genotypes of the two parents to be dependent on one another. In the recursive genotypic peeling algorithm (Fernando et al., 1993), full sibs are peeled by summing over all possible genotypes of both the sire and the dam.

Computational Considerations

The major advantage of allelic peeling compared to genotypic peeling is the scalability with increasing number of alleles. Table 3 shows that allelic peeling computations are proportional to the number of alleles squared, whereas genotypic peeling is proportional to a mixture of the number of alleles raised to powers of two, six, and eight. The memory required to compute genotype distributions of all individuals in the pedigree is proportional to the number of alleles in allelic peeling and the number of alleles squared in genotypic peeling. The number of computations and memory requirements are roughly proportional to the number of individuals in the pedigree in both allelic and genotypic peeling.

Part of the difference in number of computations results from genotypic peeling calculating full-sib probabilities exactly. The number of calculations for individuals with full sibs is proportional to the number of alleles raised to the eighth power. Although the recursive algorithm for allelic peeling does not work for pedigrees that include full sibs, an iterative algorithm for allelic peeling (Thallman et al., 2001) does not require additional computations for full sibs as compared to the same size pedigree with half sibs.

Assumptions of Conditional Independence

Genotypes of individuals that are connected may be statistically dependent on one another. For example, in Figure 1, a change in the phenotype of b could change inferences about g_k, because it could change inferences about g_i. The statistical independence of various subsets of the pedigree is a critical assumption in all of the formulas used in peeling. In Figure 1, conditioning on g_i allows the pedigree to be divided into independent subsets. For example, conditional on $g_i = [3, 2]$, a change in the phenotype of b could no longer change inferences about g_k, because their only connection is through g_i, which is now fixed. Therefore, the pedigree is divided into several disjoint subsets that are mutually independent conditional on g_i. This conditional independence of subsets of phenotypes is used in the derivations of Eq. [1] to [4] to partition the likelihoods into components that are amenable to recursion.

Relationship Between Terms in Genotypic and Allelic Peeling

The elements of the genotype distribution, $\mathbf{G}(i)$, are equivalent to the genotype probabilities of Fernando et al. (1993), $\Pr(u_i \mid y)$, provided there are no full sibs. The genotypic posterior probabilities in Fernando et al. (1993), $p_{ij}(u_i)$, are proportional to the elements of the allelic peeling expression,

$$\prod_{\substack{t \in \text{progeny}(i) \\ t \notin \text{progeny}(j)}}^{\circ} [0.5 \cdot \mathbf{L}(ti) \cdot \mathbf{1}' + 0.5 \cdot \mathbf{1} \cdot \mathbf{L}(ti)']$$

When they are scaled to sum to one, the genotypic anterior probabilities in Fernando et al. (1993), $a_i(u_i)$, are analogous to the elements of the allelic peeling expression, $\mathbf{P}(id) \cdot \mathbf{P}(is)'$.

Allelic peeling has computational advantages relative to genotypic peeling, especially for loci with many alleles. This paper establishes the framework and notation for an iterative algorithm that can handle pedigrees with loops (including full sibs) efficiently (Thallman et al., 2001).

Implications

Allelic peeling is a method for calculating genotype probabilities. For loci with many alleles, allelic peeling is much more computationally efficient than conventional peeling algorithms. Allelic peeling is especially appropriate for computing genotype probabilities of microsatellites. The method is a starting point for addressing several complicating factors commonly found in livestock pedigrees.

Literature Cited

Cannings, C., E. A. Thompson, and M. H. Skolnick. 1978. Probability functions on complex pedigrees. Adv. Appl. Probab. 10:26–61.

Elston, R. C., and J. Stewart. 1971. A general model for the genetic analysis of pedigree data. Hum. Hered. 21:523–542.

Fernando, R. L., C. Stricker, and R. C. Elston. 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, S. W., and E. A. Thompson. 1992. A Monte Carlo method for combined segregation and linkage analysis. Am. J. Hum. Genet. 51:1111–1126.

Kerr, R. J., and B. P. Kinghorn. 1996. An efficient algorithm for segregation analysis in large populations. J. Anim. Breed. Genet. 113:457–469.

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

Lincoln, S. E., and E. S. Lander. 1992. Systematic detection of errors in genetic linkage data. Genomics 14:604–610.

Ott, J. 1999. Analysis of Human Genetic Linkage. 3rd ed. Johns Hopkins University Press, Baltimore, MD.

Rohrer, G. A., and J. W. Keele. 1998. Identification of quantitative trait loci affecting carcass composition in swine: I. Fat deposition traits. J. Anim. Sci. 76:2247–2254.

Stone, R. T., J. W. Keele, S. D. Shackelford, S. M. Kappes, and M. Koohmaraie. 1999. A primary screen of the bovine genome for quantitative trait loci affecting carcass and growth traits. J. Anim. Sci. 77:1379–1384.

Thallman, R. M., G. L. Bennett, J. W. Keele, and S. M. Kappes. 2001. Efficient computation of genotype probabilities for loci with many alleles: II. Iterative method for large, complex pedigrees. J. Anim. Sci. 79:34–44.

Uimari, P., G. Thaller, and I. Hoeschele. 1996. The use of multiple markers in a Bayesian method for mapping quantitative trait loci. Genetics 143:1831–1842.

van Arendonk, J. A. M., C. Smith, and B. W. Kennedy. 1989. Method to estimate genotype probabilities at individual loci in farm livestock. Theor. Appl. Genet. 78:735–740.

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

Zhang, Q., D. Boichard, I. Hoeschele, C. Ernst, A. Eggen, B. Murkve, M. Pfister-Genskow, L. A. Witte, F. E. Grignola, P. Uimari, G. Thaller, and M. D. Bishop. 1998. Mapping quantitative trait loci for milk production and health of dairy cattle in a large outbred pedigree. Genetics 149:1959–1973.