

A rapid method for computing the inverse of the gametic covariance matrix between relatives for a marked Quantitative Trait Locus

Gamal ABDEL-AZIM*, Albert E. FREEMAN

Department of Animal Science, Iowa State University, Ames, IA 50011, USA

(Received 23 December 1999; accepted 15 November 2000)

Abstract – The inverse of the gametic covariance matrix between relatives, \mathbf{G}^{-1} , for a marked quantitative trait locus (QTL) is required in best linear unbiased prediction (BLUP) of breeding values if marker data are available on a QTL. A rapid method for computing the inverse of a gametic relationship matrix for a marked QTL without building \mathbf{G} itself is presented. The algorithm is particularly useful due to the approach taken in computing inbreeding coefficients by having to compute only few elements of \mathbf{G} . Numerical techniques for determining, storing, and computing the required elements of \mathbf{G} and the nonzero elements of the inverse are discussed. We show that the subset of \mathbf{G} required for computing the inbreeding coefficients and hence the inverse is a tiny proportion of the whole matrix and can be easily stored in computer memory using sparse matrix storage techniques. We also introduce an algorithm to determine the maximum set of nonzero elements that can be found in \mathbf{G}^{-1} and a strategy to efficiently store and access them. Finally, we demonstrate that the inverse can be efficiently built using the present techniques for very large and inbred populations.

gametic relationship / marker-assisted selection / best linear unbiased / prediction

1. INTRODUCTION

The utilization of marker quantitative trait loci associations in genetic evaluation is now possible and likely to be used more extensively in the future. Also, many authors have estimated gain through marker-assisted selection, *e.g.* [6, 7, 9, 14].

Marker information will not replace phenotypic records because a full prediction of phenotype from DNA sequence is still far from achievable [3]. Joint utilization of marker and phenotype information in current genetic prediction models is, however, progressing at a rapid pace. Fernando and Grossman [2]

* Correspondence and reprints
E-mail: gaazim@iastate.edu

explained how genetic markers associated with quantitative trait loci (QTLs) could be incorporated into mixed models. Marked QTL alleles were considered random in the context of the mixed model terminology, and algorithms to construct and invert the covariance matrix pertaining to QTL additive effects were developed.

Based on previous developments by Fernando and Grossman [2] and van Arendonk *et al.* [12], and using the partitioned matrix theory, Wang *et al.* [13] described an exact recursive method to obtain the inverse of the covariance matrix of the additive effects of a marked QTL in the case of complete marker data. If inbreeding is considered, certain elements of \mathbf{G} are required, however, Wang *et al.* [13] did not specify how these elements could be computed separately.

The objective of the present paper is to develop a rapid method to obtain the inverse of the covariance matrix of the additive effects of a marked QTL in the case of complete marker data using a small subset of \mathbf{G} . In addition to the partitioned matrix theory, we show that the inverse can also be obtained by factorizing the covariance matrix into \mathbf{LDL}' where \mathbf{L} is a lower triangular matrix whose inverse can be directly computed from pedigree and marker data. Matrix \mathbf{D} is shown to be proportional to the covariance matrix of Mendelian sampling at the QTL for given observed marker genotypes. We will show that \mathbf{D} is block diagonal and can be computed from a small subset of \mathbf{G} . The method is inspired by the rapid method of Henderson [4] to obtain the inverse of the numerator relationship matrix. In this work we will give special attention to computing efficiency. Numerical techniques to efficiently compute and store a subset of the covariance matrix and the nonzero elements of the inverse are discussed.

2. TABULAR METHODS FOR THE COVARIANCE MATRIX AND THE INVERSE

The covariance of marked QTL (MQTL) effects for given complete marker data was discussed by Fernando and Grossman [2], van Arendonk *et al.* [12], and Wang *et al.* [13]. The covariance can be divided into two parts: between individuals and within individuals. By definition, the genetic covariance between two alleles is the probability that they are identical by descent, multiplied by the additive genetic variance, σ_v^2 . In animals, each locus consists of two alleles, hence for given known marker genotypes, four covariance values can be computed between each two individuals as described in definition (1). Also, within every individual, four covariance values can be computed as described in definition (2). Denote the two MQTL alleles of individual i by α_i^1 and α_i^2 ; in addition, denote the additive effects of the two MQTL alleles of individual i by \mathbf{v}_i , where $\mathbf{v}_i = [v_i^1 v_i^2]'$. Also, let $P(\alpha_i \equiv \alpha_j | \mathbf{M})$ denote the probability that

any two alleles, say α_i and α_j , are identical by descent given \mathbf{M} with \mathbf{M} defined as the event of observing marker genotypes, then

$$\text{Cov}(\mathbf{v}_i, \mathbf{v}'_j | \mathbf{M}) = \sigma_v^2 \begin{bmatrix} P(\alpha_i^1 \equiv \alpha_j^1 | \mathbf{M}) & P(\alpha_i^1 \equiv \alpha_j^2 | \mathbf{M}) \\ P(\alpha_i^2 \equiv \alpha_j^1 | \mathbf{M}) & P(\alpha_i^2 \equiv \alpha_j^2 | \mathbf{M}) \end{bmatrix} \quad (1)$$

$$\text{Cov}(\mathbf{v}_i, \mathbf{v}'_i | \mathbf{M}) = \sigma_v^2 \begin{bmatrix} P(\alpha_i^1 \equiv \alpha_i^1 | \mathbf{M}) & P(\alpha_i^1 \equiv \alpha_i^2 | \mathbf{M}) \\ P(\alpha_i^2 \equiv \alpha_i^1 | \mathbf{M}) & P(\alpha_i^2 \equiv \alpha_i^2 | \mathbf{M}) \end{bmatrix} = \sigma_v^2 \begin{bmatrix} 1 & f_i \\ f_i & 1 \end{bmatrix}. \quad (2)$$

In definition (2) the probability of identity by descent between an allele and itself, for \mathbf{M} equals 1, and f_i the probability that the two MQTL alleles of individual i are identical by descent for \mathbf{M} ; f will be referred to as the inbreeding coefficient.

If animals are ordered such that parents precede their progeny and are identified by integers from 1 to n , then a number of n^2 covariance matrices of order 2, described in (1) and (2), can be put together in a matrix of order $2n$ that is referred to as the conditional gametic relationship matrix for given marker data [13]. Denote the element located in row r and column c of any matrix \mathbf{A} by $\mathbf{A}(r, c)$, and denote the entire r th row of \mathbf{A} by $\mathbf{A}(r, \cdot)$ and the entire c th column of \mathbf{A} by $\mathbf{A}(\cdot, c)$, then $\text{Cov}(\mathbf{v}_i, \mathbf{v}_j | \mathbf{M}) / \sigma_v^2$, and $\text{Cov}(\mathbf{v}_i, \mathbf{v}_i | \mathbf{M}) / \sigma_v^2$, that we will refer to as \mathbf{C}_{ij} and \mathbf{C}_{ii} , can be written as

$$\mathbf{C}_{ij} = \begin{bmatrix} \mathbf{G}(2i-1, 2j-1) & \mathbf{G}(2i-1, 2j) \\ \mathbf{G}(2i, 2j-1) & \mathbf{G}(2i, 2j) \end{bmatrix} \quad (3)$$

and

$$\mathbf{C}_{ii} = \begin{bmatrix} 1 & \mathbf{G}(2i-1, 2i) \\ \mathbf{G}(2i, 2i-1) & 1 \end{bmatrix}. \quad (4)$$

For example, the (1,1) element of \mathbf{C}_{ij} is the element of \mathbf{G} located in the $(2i-1)$ th row and the $(2j-1)$ th column. Because \mathbf{G} is symmetric and \mathbf{C}_{ij} is not a scalar, $\mathbf{C}_{ji} = (\mathbf{C}_{ij})'$. Moreover, all diagonal elements of \mathbf{G} are equal to 1.

2.1. Tabular method for \mathbf{G}

It can be shown that \mathbf{C}_{ij} can be computed from previous rows of \mathbf{G} [13]. For $i > j$,

$$\mathbf{C}_{ij} = \mathbf{Q}_i \begin{bmatrix} \mathbf{C}_{sj} \\ \mathbf{C}_{dj} \end{bmatrix}. \quad (5)$$

Where s and d denote paternal and maternal parents, respectively, of individual i , \mathbf{Q}_i is a 2×4 matrix defined as

$$\mathbf{Q}_i = \begin{bmatrix} P(\alpha_i^1 \leftarrow \alpha_s^1 | \mathbf{M}) & P(\alpha_i^1 \leftarrow \alpha_s^2 | \mathbf{M}) & P(\alpha_i^1 \leftarrow \alpha_d^1 | \mathbf{M}) & P(\alpha_i^1 \leftarrow \alpha_d^2 | \mathbf{M}) \\ P(\alpha_i^2 \leftarrow \alpha_s^1 | \mathbf{M}) & P(\alpha_i^2 \leftarrow \alpha_s^2 | \mathbf{M}) & P(\alpha_i^2 \leftarrow \alpha_d^1 | \mathbf{M}) & P(\alpha_i^2 \leftarrow \alpha_d^2 | \mathbf{M}) \end{bmatrix}. \quad (6)$$

The matrix \mathbf{Q}_i contains the probabilities that the paternal and maternal alleles of individual i descended from any of the four alleles of its two parents for given observed marker genotypes. Due to the marker-QTL association, the probability that an individual received the QTL allele that was in coupling phase in the parent with the marker allele it received from that parent is $1 - r$ where r is the recombination rate between the marker locus and the QTL. Based on this simple genetic fact, Wang *et al.* [13] computed \mathbf{Q}_i for the i th individual. Matrix \mathbf{Q}_i is required for each individual in the pedigree and hence its computing cost needs to be minimized. We present a general algorithm to efficiently compute \mathbf{Q}_i in Appendix A.

Because the relationship \mathbf{C}_{ij} , between the two individuals i and j , at the QTL, can be computed from already built relationships, *i.e.*, \mathbf{C}_{sj} and \mathbf{C}_{dj} as shown in (5), there exists a recursive method to build new relationships from previous elements of \mathbf{G} . The following formulation, as suggested by Wang *et al.* [13], adds the two rows corresponding to the i th individual to the lower triangle of \mathbf{G} , and using the symmetry of \mathbf{G} , the corresponding upper triangle elements are constructed:

$$\mathbf{G}_i = \begin{bmatrix} \mathbf{G}_{i-1} & \mathbf{G}_{i-1}\mathbf{A}'_i \\ \mathbf{A}_i\mathbf{G}_{i-1} & \mathbf{C}_{ii} \end{bmatrix} \quad (7)$$

where \mathbf{A}_i is a $2 \times 2(i - 1)$ matrix constructed by setting $\mathbf{A}(, 2s - 1)$ equal to $\mathbf{Q}(, 1)$, $\mathbf{A}(, 2s)$ equal to $\mathbf{Q}(, 2)$, $\mathbf{A}(, 2d - 1)$ equal to $\mathbf{Q}(, 3)$, and $\mathbf{A}(, 2d)$ equal to $\mathbf{Q}(, 4)$, the rest of \mathbf{A} is set equal to 0. The matrix \mathbf{Q} is defined in (6), and \mathbf{C}_{ii} is defined in (2). The inbreeding coefficient, f_i , is the only element required to construct \mathbf{C}_{ii} and can be computed as described in Wang *et al.* [13]. It is important for future use to know that f_i is a function of \mathbf{Q}_i , \mathbf{C}_{ss} , \mathbf{C}_{dd} , and \mathbf{C}_{sd} . Given observed marker genotypes and the recombination rate of 0.1, the conditional gametic relationship matrix for the pedigree listed in Table I is shown in Figure 1.

2.2. Decomposing \mathbf{G}

In this section we decompose \mathbf{G} following arguments similar to those Henderson [4] used in decomposing the numerator relationship matrix (NRM). The matrix \mathbf{G} can be decomposed and written as

$$\mathbf{G} = \mathbf{LDL}' \quad (8)$$

where \mathbf{L} is a lower triangular matrix and \mathbf{D} is a block diagonal matrix. Matrix \mathbf{L} can be recursively computed using relationship (9) that adds the two rows corresponding to individual i , *i.e.*, rows $2i - 1$ and $2i$ to \mathbf{L} ,

$$\mathbf{L}_i = \begin{bmatrix} \mathbf{L}_{i-1} & \mathbf{0} \\ \mathbf{A}_i\mathbf{L}_{i-1} & \mathbf{I}_2 \end{bmatrix} \quad (9)$$

Table I. Example pedigree and the corresponding \mathbf{Q}_i and \mathbf{d}_i matrices.

Animal	Sire	Dam	Genotype	\mathbf{Q}_i				\mathbf{d}_i	
1	0	0	A_1A_1	-				-	
2	0	0	A_2A_2	-				-	
3	0	0	A_1A_2	-				-	
4	1	2	A_1A_2	0.50	0.50	0.00	0.00	0.500	0.000
				0.00	0.00	0.50	0.50	0.000	0.500
5	3	4	A_1A_1	0.45	0.05	0.45	0.05	0.590	-0.410
				0.45	0.05	0.45	0.05	-0.410	0.590
6	1	4	A_1A_2	0.50	0.50	0.00	0.00	0.500	0.000
				0.00	0.00	0.10	0.90	0.000	0.180
7	5	6	A_1A_2	0.50	0.50	0.00	0.00	0.500	0.000
				0.00	0.00	0.10	0.90	0.000	0.171

	v_1^1	v_1^2	v_2^1	v_2^2	v_3^1	v_3^2	v_4^1	v_4^2	v_5^1	v_5^2	v_6^1	v_6^2	v_7^1	v_7^2
v_1^1	1.000	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.225	0.225	0.500	0.050	0.225	0.095
v_1^2	0.000	1.000	0.000	0.000	0.000	0.000	0.500	0.000	0.225	0.225	0.500	0.050	0.225	0.095
v_2^1	0.000	0.000	1.000	0.000	0.000	0.000	0.500	0.025	0.025	0.000	0.450	0.025	0.405	0.405
v_2^2	0.000	0.000	0.000	1.000	0.000	0.000	0.500	0.025	0.025	0.000	0.450	0.025	0.405	0.405
v_3^1	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.450	0.450	0.000	0.000	0.450	0.000
v_3^2	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.050	0.050	0.000	0.000	0.050	0.000
v_4^1	0.500	0.500	0.000	0.000	0.000	0.000	1.000	0.000	0.450	0.450	0.500	0.100	0.450	0.140
v_4^2	0.000	0.000	0.500	0.500	0.000	0.000	0.000	1.000	0.050	0.050	0.000	0.900	0.050	0.810
v_5^1	0.225	0.225	0.025	0.025	0.450	0.050	0.450	0.050	1.000	0.000	0.225	0.090	0.500	0.104
v_5^2	0.225	0.225	0.025	0.025	0.450	0.050	0.450	0.050	0.000	1.000	0.225	0.090	0.500	0.104
v_6^1	0.500	0.500	0.000	0.000	0.000	0.000	0.500	0.000	0.225	0.225	1.000	0.050	0.225	0.145
v_6^2	0.050	0.050	0.450	0.450	0.000	0.000	0.100	0.900	0.090	0.090	0.050	1.000	0.090	0.905
v_7^1	0.225	0.225	0.025	0.025	0.450	0.050	0.450	0.050	0.500	0.500	0.225	0.090	1.000	0.104
v_7^2	0.095	0.095	0.405	0.405	0.000	0.000	0.140	0.810	0.104	0.104	0.145	0.905	0.104	1.000

Figure 1. Conditional gametic relationship matrix.

where \mathbf{I}_2 is an identity matrix of order 2; \mathbf{A}_i is defined in (7). The relationship (9) indicates that the smallest unit of \mathbf{L} that can be built is a matrix of order 2, not a scalar as in the decomposed NRM. To illustrate this and subsequent computations, we use the pedigree of Table I. The matrix \mathbf{L} is shown in Figure 2.

To illustrate the procedure of building \mathbf{L} , denote the 2×2 matrix on the intersection of the two individuals i and j by $\mathbf{R}(i, j)$. Now given that 5 and 6

	v_1^1	v_1^2	v_2^1	v_2^2	v_3^1	v_3^2	v_4^1	v_4^2	v_5^1	v_5^2	v_6^1	v_6^2	v_7^1	v_7^2
v_1^1	1.000													
v_1^2	0.000	1.000												
v_2^1	0.000	0.000	1.000											
v_2^2	0.000	0.000	0.000	1.000										
v_3^1	0.000	0.000	0.000	0.000	1.000									
v_3^2	0.000	0.000	0.000	0.000	0.000	1.000								
v_4^1	0.500	0.500	0.000	0.000	0.000	0.000	1.000							
v_4^2	0.000	0.000	0.500	0.500	0.000	0.000	0.000	1.000						
v_5^1	0.225	0.225	0.025	0.025	0.450	0.050	0.450	0.050	1.000					
v_5^2	0.225	0.225	0.025	0.025	0.450	0.050	0.450	0.050	0.000	1.000				
v_6^1	0.500	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000			
v_6^2	0.050	0.050	0.450	0.450	0.000	0.000	0.100	0.900	0.000	0.000	0.000	1.000		
v_7^1	0.225	0.225	0.025	0.025	0.450	0.050	0.450	0.050	0.500	0.500	0.000	0.000	1.000	
v_7^2	0.095	0.095	0.405	0.405	0.000	0.000	0.090	0.810	0.000	0.000	0.100	0.900	0.000	1.000

Figure 2. L matrix.

are parents of 7, we compute $\mathbf{R}(7, 2)$ as

$$\begin{aligned} \mathbf{R}(7, 2) &= \mathbf{Q}_7 \begin{bmatrix} \mathbf{R}(5, 2) \\ \mathbf{R}(6, 2) \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.1 & 0.9 \end{bmatrix} \begin{bmatrix} 0.025 & 0.025 \\ 0.025 & 0.025 \\ 0 & 0 \\ 0.45 & 0.45 \end{bmatrix} \\ &= \begin{bmatrix} 0.025 & 0.025 \\ 0.405 & 0.405 \end{bmatrix}. \end{aligned}$$

The variance and covariance of Mendelian sampling for an individual with two alleles at the marked QTL for given observed marker genotypes is described by a matrix of order 2, say \mathbf{d}_i , and not by a scalar as in the case of the infinitesimal model. Define \mathbf{D} as $\text{diag}[\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n]$, where the 2×2 \mathbf{d}_i can be computed separately for each individual. It will be shown that $\sigma_v^2 \mathbf{D}$ is the covariance matrix of Mendelian sampling due to a QTL linked to one marker for given observed marker genotypes. To find the conditional Mendelian sampling covariance for individual i , denote its Mendelian sampling by \mathbf{m}_i , then

$$\mathbf{m}_i = \mathbf{v}_i - \mathbf{Q}_i \begin{bmatrix} \mathbf{v}_s \\ \mathbf{v}_d \end{bmatrix}. \quad (10)$$

The rationale behind (10) is that Mendelian sampling in the case of a marked QTL could be computed just as in the case of the infinitesimal model, by

subtracting the expected breeding value from the realized breeding value. It can now be proved that

$$\frac{1}{\sigma_v^2} \text{Var}(\mathbf{m}_i | \mathbf{M}) = \mathbf{C}_{ii} - \mathbf{Q}_i \begin{bmatrix} \mathbf{C}_{ss} & \mathbf{C}_{sd} \\ \mathbf{C}_{ds} & \mathbf{C}_{dd} \end{bmatrix} \mathbf{Q}'_i. \quad (11)$$

See Appendix B for a proof of (11). Further, for a proof that \mathbf{D} is block diagonal, see Appendix C. From (11), \mathbf{d}_i can be computed as

$$\mathbf{d}_i = \mathbf{C}_{ii} - \mathbf{Q}_i \begin{bmatrix} \mathbf{C}_{ss} & \mathbf{C}_{sd} \\ \mathbf{C}_{ds} & \mathbf{C}_{dd} \end{bmatrix} \mathbf{Q}'_i. \quad (12)$$

To illustrate (12), we compute \mathbf{d} for individual 7,

$$\begin{aligned} \mathbf{d}_7 &= \begin{bmatrix} 1 & 0.104 \\ 0.104 & 1 \end{bmatrix} \\ &\quad - \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.1 & 0.9 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0.225 & 0.09 \\ 0 & 1 & 0.225 & 0.09 \\ 0.225 & 0.225 & 1 & 0.05 \\ 0.09 & 0.09 & 0.05 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0 \\ 0.5 & 0 \\ 0 & 0.1 \\ 0 & 0.9 \end{bmatrix} \\ &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.17 \end{bmatrix}. \end{aligned}$$

Now it is straightforward to verify the decomposition of \mathbf{G} by the direct multiplication \mathbf{LDL}' .

2.3. Computing the inverse of \mathbf{G}

The inverse of \mathbf{G} is now computed by making use of the decomposition presented earlier. From the decomposition of \mathbf{G} in (8), \mathbf{G}^{-1} can be written as

$$\mathbf{G}^{-1} = (\mathbf{L}')^{-1} \mathbf{D}^{-1} \mathbf{L}^{-1}. \quad (13)$$

\mathbf{L}^{-1} is easy to compute due to the recursive method used to construct \mathbf{L} . The inverse of \mathbf{L} can be computed according to the following recursive relationship

$$\mathbf{L}_i^{-1} = \begin{bmatrix} \mathbf{L}_{i-1}^{-1} & \mathbf{0} \\ -\mathbf{A}_i & \mathbf{I}_2 \end{bmatrix} \quad (14)$$

as can be verified by showing that

$$\begin{bmatrix} \mathbf{L}_{i-1}^{-1} & \mathbf{0} \\ -\mathbf{A}_i & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{L}_{i-1} & \mathbf{0} \\ \mathbf{A}_i \mathbf{L}_{i-1} & \mathbf{I}_2 \end{bmatrix} = \mathbf{I}.$$

Using the recursive relationship (14), \mathbf{L}^{-1} for the pedigree of Table I is

	v_1^1	v_1^2	v_2^1	v_2^2	v_3^1	v_3^2	v_4^1	v_4^2	v_5^1	v_5^2	v_6^1	v_6^2	v_7^1	v_7^2
v_1^1	1.000													
v_1^2	0.000	1.000												
v_2^1	0.000	0.000	1.000											
v_2^2	0.000	0.000	0.000	1.000										
v_3^1	0.000	0.000	0.000	0.000	1.000									
v_3^2	0.000	0.000	0.000	0.000	0.000	1.000								
v_4^1	-0.500	-0.500	0.000	0.000	0.000	0.000	1.000							
v_4^2	0.000	0.000	-0.500	-0.500	0.000	0.000	0.000	1.000						
v_5^1	0.000	0.000	0.000	0.000	-0.450	-0.050	-0.450	-0.050	1.000					
v_5^2	0.000	0.000	0.000	0.000	-0.450	-0.050	-0.450	-0.050	0.000	1.000				
v_6^1	-0.500	-0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000			
v_6^2	0.000	0.000	0.000	0.000	0.000	0.000	-0.100	-0.900	0.000	0.000	0.000	1.000		
v_7^1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.500	-0.500	0.000	0.000	1.000	
v_7^2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.100	-0.900	0.000	1.000

As mentioned before, \mathbf{D} is composed of the \mathbf{d}_i matrices along its diagonal, and \mathbf{d}_i is proportional to the conditional variance and covariance of Mendelian sampling within individual i . Therefore, \mathbf{d}_i is positive definite and can be written as

$$\mathbf{d}_i = \mathbf{t}_i \mathbf{t}_i' \quad (15)$$

where \mathbf{t}_i is a matrix of order 2 with 0 as its upper diagonal element. If \mathbf{d}_i is symbolically written as

$$\mathbf{d}_i = \begin{bmatrix} p & k \\ k & q \end{bmatrix} \quad \text{and} \quad c = \sqrt{q - k^2/p},$$

then

$$\mathbf{t}_i^{-1} = \begin{bmatrix} 1/\sqrt{p} & 0 \\ -k/pc & 1/c \end{bmatrix} \quad (16)$$

where p , k , q , and c are scalars. Notice that p and q are the conditional Mendelian sampling variances associated with α_i^1 and α_i^2 , respectively, and k is their conditional Mendelian sampling covariance. The results in (16) can be easily seen by inverting \mathbf{t}_i , obtained after the decomposition of \mathbf{d}_i in (15). Relationship (16) shows an easy way to compute \mathbf{t}_i^{-1} directly from elements of \mathbf{d}_i without having to decompose \mathbf{d}_i and then invert \mathbf{t}_i .

After every \mathbf{d}_i is decomposed as described in (15), \mathbf{D} can be written as $\mathbf{T}\mathbf{T}'$ where \mathbf{T} is lower triangular defined as $\text{diag}[\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_n]$. The matrix \mathbf{D}^{-1} can then be written as $(\mathbf{T}')^{-1}\mathbf{T}^{-1}$ and

$$\mathbf{G}^{-1} = (\mathbf{L}')^{-1}(\mathbf{T}')^{-1}\mathbf{T}^{-1}\mathbf{L}^{-1}. \quad (17)$$

Since the inverse has the form (17), the contribution of each individual to \mathbf{G}^{-1} is now easy to compute using the recursive method for constructing \mathbf{L}^{-1} and the efficient way of expression (16) that can be used to directly obtain \mathbf{t}_i^{-1} from the elements of \mathbf{d}_i . From (17), the contribution of the i th individual to the inverse can be written as the cross product of $(\mathbf{t}_i^{-1}(-\mathbf{A}_i\mathbf{I}_2))$, where the cross product of any matrix, say \mathbf{B} , is $\mathbf{B}'\mathbf{B}$. Since the nonzero elements of \mathbf{A}_i are the elements of the matrix \mathbf{Q}_i , the cross product of

$$(\mathbf{t}_i^{-1}[-\mathbf{Q}_i\mathbf{I}_2]) \quad (18)$$

is added to the following locations of \mathbf{G}^{-1} ,

$$\begin{bmatrix} \mathbf{R}(s, s) & \mathbf{R}(s, d) & \mathbf{R}(s, i) \\ \mathbf{R}(d, s) & \mathbf{R}(d, d) & \mathbf{R}(d, i) \\ \mathbf{R}(i, s) & \mathbf{R}(i, d) & \mathbf{R}(i, i) \end{bmatrix}, \quad (19)$$

where i, s, d , and $\mathbf{R}(i, j)$ are consistent with their previous definitions, with $\mathbf{R}(i, s)$ for example, as the matrix of order 2 at the intersection of the individual and its paternal parent.

2.4. Algorithm

Next, we suggest an algorithm to compute and add the contributions of the i th individual to \mathbf{G}^{-1} .

- **Set** a 2×6 matrix, say $\mathbf{\Delta}$ to $\mathbf{0}$.
- **Set** elements 1 to 6 of a 6×1 vector, say τ , to $2s - 1, 2s, 2d - 1, 2d, 2i - 1$, and $2i$, in order.
- **Compute** \mathbf{d}_i as described in (12) and **assign** $1/\sqrt{p}$ to $\mathbf{\Delta}(1, 5)$, 0 to $\mathbf{\Delta}(1, 6)$, $-k/pc$ to $\mathbf{\Delta}(2, 5)$, and $1/c$ to $\mathbf{\Delta}(2, 6)$.
- **Assign** $-\mathbf{Q}(1,)/\sqrt{p}$ to elements 1 to 4 of $\mathbf{\Delta}(1,)$.
- **Assign** $(k\mathbf{Q}(1,)/pc - \mathbf{Q}(2,)/c)$ to elements 1 to 4 of $\mathbf{\Delta}(2,)$.
- **For** $x = 1$ to 6
 For $y = 1$ to 6
 Add $(\mathbf{\Delta}(1, x)\mathbf{\Delta}(1, y) + \mathbf{\Delta}(2, x)\mathbf{\Delta}(2, y))$ to $\mathbf{G}^{-1}(\tau(x), \tau(y))$.

The algorithm does not explicitly invert or decompose \mathbf{D} , it only computes the elements of \mathbf{t}_i^{-1} according to (16) after computing \mathbf{d}_i for each individual. Furthermore, instead of carrying out the matrix product of (18), the algorithm directly assigns the multiplication results to the 2×6 matrix $\mathbf{\Delta}$.

To illustrate the computation of $\mathbf{\Delta}$, we compute $\mathbf{\Delta}_7$. From \mathbf{d}_7 of Table I, $c = \sqrt{0.171 - 0/0.5} = 0.413$, $\sqrt{m} = 0.707$, and hence

$$\mathbf{\Delta}_7 = \begin{bmatrix} -0.707 & -0.707 & 0 & 0 & 1.414 & 0 \\ 0 & 0 & -0.242 & -2.176 & 0 & 2.418 \end{bmatrix}.$$

The matrix \mathbf{G}^{-1} for the example follows:

	v_1^1	v_1^2	v_2^1	v_2^2	v_3^1	v_3^2	v_4^1	v_4^2	v_5^1	v_5^2	v_6^1	v_6^2	v_7^1	v_7^2
v_1^1	2.000	1.000	0.000	0.000	0.000	0.000	-1.000	0.000	0.000	0.000	-1.000	0.000	0.000	0.000
v_1^2	1.000	2.000	0.000	0.000	0.000	0.000	-1.000	0.000	0.000	0.000	-1.000	0.000	0.000	0.000
v_2^1	0.000	0.000	1.500	0.500	0.000	0.000	0.000	-1.000	0.000	0.000	0.000	0.000	0.000	0.000
v_2^2	0.000	0.000	0.500	1.500	0.000	0.000	0.000	-1.000	0.000	0.000	0.000	0.000	0.000	0.000
v_3^1	0.000	0.000	0.000	0.000	3.250	0.250	2.250	0.250	-2.500	-2.500	0.000	0.000	0.000	0.000
v_3^2	0.000	0.000	0.000	0.000	0.250	1.028	0.250	0.028	-0.278	-0.278	0.000	0.000	0.000	0.000
v_4^1	-1.000	-1.000	0.000	0.000	2.250	0.250	4.306	0.750	-2.500	-2.500	0.000	-0.556	0.000	0.000
v_4^2	0.000	0.000	-1.000	-1.000	0.250	0.028	0.750	6.528	-0.278	-0.278	0.000	-5.000	0.000	0.000
v_5^1	0.000	0.000	0.000	0.000	-2.500	-0.278	-2.500	-0.278	3.778	2.778	0.000	0.000	-1.000	0.000
v_5^2	0.000	0.000	0.000	0.000	-2.500	-0.278	-2.500	-0.278	2.778	3.778	0.000	0.000	-1.000	0.000
v_6^1	-1.000	-1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	2.058	0.526	0.000	-0.585
v_6^2	0.000	0.000	0.000	0.000	0.000	0.000	-0.556	-5.000	0.000	0.000	0.526	10.292	0.000	-5.263
v_7^1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-1.000	-1.000	0.000	0.000	2.000	0.000
v_7^2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.585	-5.263	0.000	5.848

2.5. One unknown parent

If one of the two parents of i is unknown, Wang *et al.* [13] have suggested replacing the two columns of \mathbf{Q}_i that belong to the unknown parent with zeros, *i.e.*, considering the probability of QTL descent from the unknown parent as undefined. This approach, however, creates unusual singularities in some cases while inverting the gametic relationship matrix. For example, if the dam is unknown and both the sire and the individual have marker genotype A_1A_2 , the contributions to the inverse due to i cannot be computed either by the current algorithm or by the Wang *et al.* [13] algorithm.

Phantom identification numbers could be assigned to the unknown parents and the problem becomes a pedigree with incomplete marker data. For incomplete marker data, alternative exact and approximate approaches are available

(Wang *et al.*, 1995). The current techniques are still useful for the case of one unidentified parent and the case of incomplete marker data in general. For instance, if d is a phantom parent of i , the most probable genotype of d for given s and i genotypes could simply be assigned to d , and approximate \mathbf{G} or \mathbf{G}^{-1} values could be built as described earlier.

3. COMPUTATIONAL TECHNIQUES FOR CONSTRUCTING THE INVERSE

In most animal breeding applications, large data are commonplace. In this case, handling matrices like \mathbf{G} and its inverse within computer memory is most unlikely. Also, having to build these matrices on disk degrades performance due to the repeated search that has to take place for certain elements of \mathbf{G} and \mathbf{G}^{-1} . In this section we explain a scheme to compute the minimum possible set of \mathbf{G} that contains elements required for computing the inverse. A sparse matrix technique to store this set is also presented. In addition, due to the sparse structure of the inverse, we suggest a method that can be used to determine the maximum possible set of the nonzero elements found in the inverse and corresponding sparse matrix techniques to efficiently store and retrieve them.

3.1. Computing a subset of \mathbf{G}

Building the inverse as described earlier requires the 2×2 blocks \mathbf{C}_{ii} and \mathbf{C}_{sd} of \mathbf{G} to be available if inbreeding is to be accounted for. As was shown by Tier [11], the diagonal of the NRM can be computed from a small subset of the matrix. Although the diagonal of \mathbf{G} is known to consist of 1s, and hence need not be computed, we will use a similar approach to compute the \mathbf{C}_{ii} submatrices located on the diagonal of \mathbf{G} . Besides, in our case, extra elements are needed for the inverse, *i.e.*, the relationship of the two parents, but this does not increase the computational task because \mathbf{C}_{sd} is needed for computing \mathbf{C}_{ii} .

For the example pedigree, the set of filled cells in Table II contains the required elements. We express the subset in terms of the \mathbf{C}_{ij} and \mathbf{C}_{ii} submatrices instead of single elements. The reason for this is its computational advantage. The subset is first determined and then computed according to equation (5) and rules explained in Wang *et al.* [13]. First, to determine the subset, read the pedigree and flag cells \mathbf{C}_{ii} and \mathbf{C}_{sd} if $s > d$ or \mathbf{C}_{ds} if $d > s$, *i.e.*, the relationship between the two parents located in the lower triangle. The cells required for computing the previously flagged cells are determined as follows: starting from the second to the last row of cells and proceeding up and to the left, flag the two cells corresponding to \mathbf{C}_{sj} and \mathbf{C}_{dj} as described in equation (5). Second, after determining all the required cells, compute them row by row starting with row 1.

Table II. The subset of the gametic relationship matrix required for building the inverse.

I_2		* ¹		*		
C_{21}	I_2	*				
C_{31}	C_{32}	I_2	*			
C_{41}		C_{43}	C_{44}	*		
C_{51}			C_{54}	C_{55}		
				C_{65}	C_{66}	
						C_{77}

¹ An asterisk indicates a cell required in the upper triangle that is taken from the lower triangle.

Constructing only the required subset of \mathbf{G} found in its lower triangle saves computational time and storage requirements. For every row of cells, \mathbf{Q}_i is built only once and used for all cells in the row. An asterisk “*” indicates a required upper triangular cell that is obtained from the lower triangle. For example, in Table II, C_{43} was flagged instead of C_{34} . In the computing step however, $(C_{43})'$ is used whenever C_{34} is required.

For this method to be useful, it is necessary to employ a sparse storage scheme that allows efficient storage and retrieval of elements of the subset. A row-linked list approach is suggested in this case for two reasons: cells in a row are not determined and flagged in any particular order, and the number of filled cells in a row is not known *a priori*. Henceforth, a row-linked list will refer to the sequence of filled lower triangular elements in a row as stored in the linked lists.

To explain the storage scheme, specify the number of filled cells by n_f and the number of individuals by n . Define the following arrays: an integer array of length n_f , **column**, containing cell column indices; an integer array of length n_f , **link**, containing pointers to the location of the next cell added to a list; a double $n_f \times 4$ array, **values**, a row of which contains the four values of an off-diagonal cell; and a double array of length n, f , containing inbreeding coefficients. Row

Table III. Linked lists of the subset of the gametic relationship matrix required for building the inverse.

i	$\mathbf{column}(i)$	$\mathbf{link}(i)$	$\mathbf{values}(i, 1)$	$\mathbf{values}(i, 2)$	$\mathbf{values}(i, 3)$	$\mathbf{values}(i, 4)$	$f(i)$
1	0	0	0.00000	0.00000	0.00000	0.00000	0.00000
2	1	0	0.00000	0.00000	0.00000	0.00000	0.00000
3	1	10	0.00000	0.00000	0.00000	0.00000	0.00000
4	1	8	0.50000	0.50000	0.00000	0.00000	0.00000
5	1	9	0.22500	0.22500	0.22500	0.22500	0.00000
6	5	0	0.22500	0.22500	0.09000	0.09000	0.05000
7	0	0	0.00000	0.00000	0.00000	0.00000	0.10350
8	3	0	0.00000	0.00000	0.00000	0.00000	
9	4	0	0.45000	0.05000	0.45000	0.05000	
10	2	0	0.00000	0.00000	0.00000	0.00000	

indices are assumed to be sorted in ascending order corresponding to the first n entries of \mathbf{column} , that is, for $i = 1$ to n , the column index of the first entry to row list i is $\mathbf{column}(i)$. A value of 0 in $\mathbf{column}(i)$ indicates no entries have yet been added to the i th list. A value of 0 in $\mathbf{link}(j)$ indicates a terminal link, that is, the last entry in a list.

Linked lists for the example are given in Table III. To add an entry, \mathbf{C}_{ij} , to the lists, start at $\mathbf{column}(i)$ where i is the individual to which the entry belongs. If $\mathbf{column}(i) = 0$, place the column index of the entry, j , in $\mathbf{column}(i)$ and the four values of \mathbf{C}_{ij} in $\mathbf{values}(i, \cdot)$, otherwise proceed *via* links to search whether the array \mathbf{column} already contains j . If not, store j and \mathbf{C}_{ij} in the next available entry of \mathbf{column} and \mathbf{values} , respectively. To add an entry, \mathbf{C}_{ii} , to the lists, only place the value of f_i in the i th element of vector f , *i.e.*, in $f(i)$.

To retrieve the \mathbf{C}_{ij} entry from the linked lists, search for the entry starting at $\mathbf{column}(i)$ and proceed *via* links until the desired column index is found, *i.e.*, j . The entries in a row list do not have to be sorted in any order because the search method we described does not require any ordering. It is likely that a better searching technique will require sorting the lists. In this case, the improved searching technique is useful only if the time saved is greater than the sorting time. Notice that in linked lists new elements are usually added to the lists by inserting them in order. This practice, when tested, consumed more time than just adding new elements to the next available entry as described earlier.

3.2. Sparse storage scheme for \mathbf{G}^{-1}

For large numbers of animals, neither \mathbf{G} nor the inverse can be handled in memory. We introduce a sparse storage scheme that allows construction of the inverse within memory. The scheme first determines a maximum set

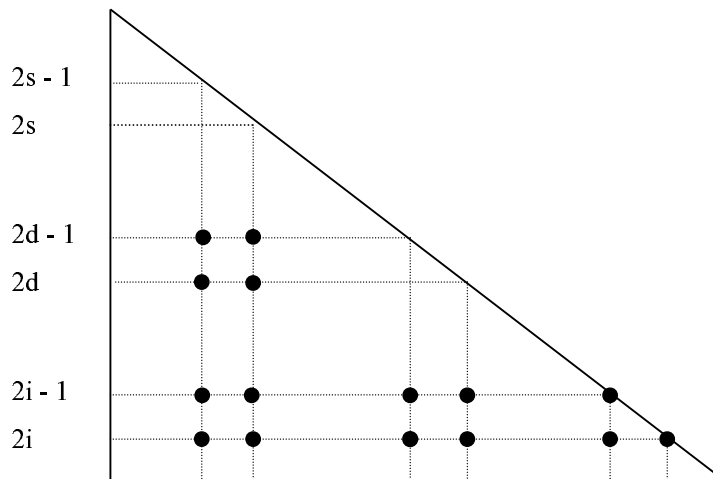


Figure 3. Contribution of an individual to the nonzero elements of the lower triangle of the gametic relationship inverse. A dark connector indicates a filled element of the inverse. In the following, $2i - 1$ and $2i$ indicate the two rows of individual i , $2d - 1$ and $2d$ indicate the two rows of the dam, and $2s - 1$ and $2s$ indicate the two rows of the sire. Perpendicular lines to the previous rows indicate the corresponding columns.

of the nonzero elements found in the lower triangle of the inverse and then computes them. The scheme is similar to that described earlier in storing and retrieving the required subset of \mathbf{G} , except that three of the four elements of the $\mathbf{R}(i, i)$ submatrices on the diagonal of \mathbf{G}^{-1} must be stored instead of only one element, f_i , of \mathbf{C}_{ii} . Only three elements are stored because of the symmetry of \mathbf{G}^{-1} . Therefore the one-dimensional array, f , is replaced by an $n \times 3$ array, say **diagv**.

Using the matrix of (19), the maximum set can be determined while reading the pedigree by adding to the lists the following entries corresponding to each individual. For the i th individual, add either $\mathbf{R}(d, s)$ or $\mathbf{R}(s, d)$ to the lists, depending on which of them is in the lower triangle; also add $\mathbf{R}(i, s)$, and $\mathbf{R}(i, d)$. Entries for $\mathbf{R}(i, i)$, $\mathbf{R}(s, s)$, and $\mathbf{R}(d, d)$ are automatically stored in the lists, in **diagv**. The sparse scheme sets an upper bound for the set of nonzero elements of the inverse that does not exceed $15n$, and its proportion out of $4n^2$, the order of \mathbf{G} , does not exceed $15/4n$. This can be seen in Figure 3. The dark connectors in the figure indicate the maximum number of filled elements that individual i could ever cause. Notice that the proportion $15/4n$ indicates that the percentage of filled elements dramatically decreases as n increases. The simulated data summarized in Table IV clearly show that as the number of individuals in the pedigree increases, the percentage of filled cells in the inverse substantially decreases.

Table IV. CPU¹ time and storage requirements for building \mathbf{G}^{-1} .

Years simulated	Number of animals	Number of required elements of \mathbf{G}	% of elements of \mathbf{G} required for the inverse	Average ² inbreeding coefficient	Number of stored elements of \mathbf{G}^{-1}	% of the elements of \mathbf{G}^{-1} stored	CPU Time ³ (s)
15	18 801	200 845	0.01420	0.06	255 327	0.018058	1.02
30	137 680	1.41963×10^6	0.00187	0.19	1.92279×10^6	0.002536	7.73
40	485 462	4.90449×10^6	0.000520	0.10	6.79909×10^6	0.000721	25.65

¹ A 433 MHz single-processor workstation running digital UNIX was used in the evaluation.

² Computed for all animals including base population.

³ This time includes the time for reading the pedigree file, setting up the lists, and computing the inverse.

After the maximum set has been determined, the algorithm described earlier can be used to compute and add contributions of the i th individual to the inverse. Because the elements of (19) have to be retrieved and added to, perhaps several times, the values of the maximum set must be first set to 0. The same search method used with \mathbf{G} is used here to retrieve the elements of the inverse. Searching *via* links is only required if it is for $\mathbf{R}(i, j)$ where $i > j$. If $i = j$, then $\mathbf{R}(i, i)$ can be directly retrieved from $\mathbf{diagv}(i,)$.

Now it should be clear that storing the elements of the matrices in groups of 4, *i.e.*, $\mathbf{R}(i, j)$, saves a great deal of computing time although it could contain zero elements. Notice that a single element of \mathbf{C}_{ij} or $\mathbf{R}(i, j)$ of the inverse is never required and hence searched for unless the other three elements are required as well. Also, no search is required if $i = j$. Although storing only the nonzero elements of the 2×2 blocks is more memory-efficient, it showed very poor performance in terms of speed when tested. More details of programming strategies can be inferred from the C code listed in Appendix C.

4. SIMULATION AND VALIDATION

In this section we use simulated pedigree and genotype data to investigate the efficiency of the algorithms. A modified nucleus scheme where sires are selected in two stages was simulated. The objective was to simulate a structural pedigree similar to what could be encountered in the U.S. Holstein population. Breeding values were simulated according to a finite locus model. A situation in which one QTL is associated with a known marker was simulated.

Data sets with variable sizes were simulated. Table IV shows that for larger data sets both the required subset of \mathbf{G} and the number of nonzero elements of the inverse constitute a tiny proportion of $4n^2$. Results of three pedigree data simulated over 15, 30, and 40 years are listed in Table IV. The first pedigree comprising 18 801 animals started with 6 active sires and 14 young bulls with a maximum of 50 daughters per young bull. We used a base cow population of 2 000 cows with a maximum of 5 lactation seasons and with culling ratios of 0.22, 0.26, 0.29, 0.34, and 1 for parities 1, 2, 3, 4, and 5, respectively. The second and third pedigrees were simulated similarly, except that the simulation was continued for 30 and 40 years, resulting in a generation of 137 680 and 485 462 animals, respectively. The percentages presented in the table are the number of physically stored single elements and not the number of the $\mathbf{R}(i, j)$ matrices. However, this number does not include the overhead caused by storing the links and column indices. The CPU seconds presented in the table indicate that by using the current algorithms, building the inverse of the conditional gametic relationship matrix for a marked QTL is as trivial as building the inverse of the NRM.

5. DISCUSSION

An algorithm to directly build the inverse of a conditional gametic relationship matrix, from given marker data, was developed. The inverse algorithm is based on matrix decomposition instead of partitioned matrix theory. Numerical techniques that greatly improved computing performance were introduced. Extension to multiple markers should be straightforward provided that MQTL loci are independent. With multiple markers, efficiency should improve relatively because column indices and link pointers are the same for all markers and could be determined and stored only once.

It is imperative to mention that although both matrix decomposition and partitioned matrix theory produce the same elements that an individual contributes to the inverse, matrix decomposition offers a more computationally useful structure to the mixed model applications. First, \mathbf{D} in our study could be used in a way similar to the Henderson \mathbf{D} in the context of the reduced animal model [1, 5, 10] to absorb the non-parental equations of the MQTL and polygenes. Moreover, careful inspection of the mechanics of building \mathbf{D} could lead to more useful reductions pertinent to the inclusion of markers in the mixed model. Reducing the number of equations is crucial if marker data are to be practically used in genetic evaluation models.

Furthermore, the decomposition allows for more flexibility in handling the mixed model equations, for instance, the Henderson \mathbf{L} was used by Quass [8] in transforming the equations in a way that could be useful for variance components estimation methods. The decomposition we introduced allows for the same technique to be adapted to handle a mixed model with markers. This is only to name some examples, but strictly speaking, wherever the factors of the decomposed numerator relationship matrix or the Mendelian sampling variance are useful, the decomposed conditional gametic relationship matrix and the conditional Mendelian sampling covariance, introduced in this study, could be exploited similarly. The algorithm should motivate further research to build on past experience for the developing area of marker-assisted selection.

ACKNOWLEDGEMENTS

Journal Paper No J-18661 of the Iowa Agriculture and Home Economics Experimental Station, Ames, Iowa. Project No. 3146, and supported by the Hatch Act and State of Iowa funds.

REFERENCES

- [1] Cantet R.J.C., Smith C., Reduced animal model for marker-assisted selection using best linear unbiased prediction, *Genet. Sel. Evol.* 23 (1991) 221–233.
- [2] Fernando R.L., Grossman M., Marker assisted selection using best linear unbiased prediction, *Genet. Sel. Evol.* 21 (1989) 467–477.

- [3] Haley C.S., Visscher P.M., Strategies to utilize marker-quantitative trait loci associations, *J. Dairy Sci.* 81 (1998) 85–97.
- [4] Henderson C.R., A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values, *Biometrics* 32 (1976) 69–83.
- [5] Henderson C.R., Theoretical basis and computational methods for a number of different animal models, *J. Dairy Sci.* 71(Suppl 2) (1988) 1–16.
- [6] Meuwissen T.H.E., Goddard M.E., The use of marker haplotypes in animal breeding schemes, *Genet. Sel. Evol.* 28 (1996) 161–176.
- [7] Meuwissen T.H.E., van Arendonk J.A.M., Potential improvement in rate of genetic gain from marker-assisted selection in dairy cattle breeding schemes, *J. Dairy Sci.* 75 (1992) 1651–1659.
- [8] Quass R.L., Transformed mixed model equations: a recursive algorithm to eliminate \mathbf{A}^{-1} , *J. Dairy Sci.* 72 (1989) 1937–1941.
- [9] Ruane J., Colleau J.J., Marker-assisted selection for a sex-limited character in a nucleus breeding population, *J. Dairy Sci.* 79 (1996) 1666–1678.
- [10] Saito S., Iwaisaki H., A reduced animal model approach to predicting total additive genetic merit for marker-assisted selection, *Genet. Sel. Evol.* 29 (1997) 25–34.
- [11] Tier B., Computing inbreeding coefficients quickly, *Genet. Sel. Evol.* 22 (1990) 419–430.
- [12] van Arendonk J.A.M., Tier B., Kinghorn B.P., Use of multiple genetic markers in prediction of breeding values, *Genetics* 137 (1994) 319–329.
- [13] Wang T., Fernando R.L., van der Beek S., Grossman M., van Arendonk J.A.M., Covariance between relatives for a marked quantitative trait locus, *Genet. Sel. Evol.* 27 (1995) 251–274.
- [14] Zhang W., Smith S., Computer simulation of marker-assisted selection utilizing linkage disequilibrium, *Theor. Appl. Genet.* 83 (1992) 813–820.

APPENDIX A

Algorithm to compute \mathbf{Q}_i

The theory for computing the probability of QTL descent based on the probability of marker descent and the recombination rate, r , given marker data, was discussed by Fernando and Grossman [2], van Arendonk *et al.* [12], and Wang *et al.* [13]. The following is a general algorithm to compute the matrix \mathbf{Q}_i for any number of alleles segregating in the population. The algorithm avoids building the intermediate matrices PDMs and R-, see Wang *et al.* [13]. The following is a C function that receives individual, sire, and dam identification numbers (i , s , and d , respectively) in addition to a pointer to the matrix containing marker allele genotypes in two columns (B). The function returns a pointer for Q after building it.

```

double **MakeQ(int i, int s, int d, int **B) {
    int l=2, c, x1, x2, k, j, g[7], o1, o2, c1, c2;
    double r = Recombination Rate;
    double **Q; Q = dmatrix(1,2,1,4); /* allocate Q */
    for(k=1;k<=2;k++) for(j=1;j<=4;j++) Q[k][j] = 0.;
    if(s==0 || d==0) return Q;

    g[1]=B[i][1]; g[2]=B[i][2]; g[3]=B[s][1];
    g[4]=B[s][2]; g[5]=B[d][1]; g[6]=B[d][2];
    for(k=1;k<=2;k++) {
        x1 = x2 = 0;
        if(g[1]==g[5] || g[1]==g[6])
            for(j=3;j<=4;j++) if(g[k]==g[j]) x1++;
        if(g[1]==g[3] || g[1]==g[4])
            for(j=5;j<=6;j++) if(g[k]==g[j]) x2++;

        if((x1+x2) == 3){ x1 *= 2; x2*=2; }
        else { c=x1; x1+=x2; x2+=c; }

        o1=5; o2=6; c1=3; c2=4;
        c = x1; /* for j = 1, c will act in place of x1 */
        for(j=1; j<=2; j++) {
            if(g[1]==g[o1] || g[1]==g[o2])
                if(g[k]==g[c1] || g[k]==g[c2]) {
                    if(g[c1]==g[c2]) Q[k][c1--2]=Q[k][c2--2]=1./c;
                    else if(g[k]==g[c1]) {Q[k][c1--2]=(1-r)/c; Q[k][c2--2]=r/c ;
                    else { Q[k][c1--2]=r/c; Q[k][c2--2]=(1-r)/c;} }
                swap(&o1, &c1); swap(&o2, &c2);
                c = x2; /* for j = 2, c will act in place of x2 */
            }
            l--;
        }
    }
    return Q;
}

```

APPENDIX B

Computing Mendelian sampling conditional covariance

From relationship (10) in the text, Mendelian sampling of the i th individual is written as

$$\mathbf{m}_i = \mathbf{v}_i - \mathbf{Q}_i \begin{bmatrix} \mathbf{v}_s \\ \mathbf{v}_d \end{bmatrix}$$

then

$$\text{Var}(\mathbf{m}_i|\mathbf{M}) = \text{Var}(\mathbf{v}_i|\mathbf{M}) + \mathbf{Q}_i \text{Var} \left(\begin{bmatrix} \mathbf{v}_s \\ \mathbf{v}_d \end{bmatrix} | \mathbf{M} \right) \mathbf{Q}_i' - 2 \text{Cov} \left(\mathbf{v}_i, \begin{bmatrix} \mathbf{v}_s \\ \mathbf{v}_d \end{bmatrix}' | \mathbf{M} \right)$$

and

$$\text{Var}(\mathbf{m}_i|\mathbf{M})/\sigma_v^2 = \mathbf{C}_{ii} + \mathbf{Q}_i \begin{bmatrix} \mathbf{C}_{ss} & \mathbf{C}_{sd} \\ \mathbf{C}_{ds} & \mathbf{C}_{dd} \end{bmatrix} \mathbf{Q}_i' - 2(\mathbf{C}_{is}\mathbf{C}_{id})\mathbf{Q}_i'. \quad (\text{B.1})$$

From (5) in the text, \mathbf{C}_{is} is computed as

$$\mathbf{C}_{is} = \mathbf{Q}_i \begin{bmatrix} \mathbf{C}_{ss} \\ \mathbf{C}_{ds} \end{bmatrix} \quad (\text{B.2})$$

and \mathbf{C}_{id} is similarly computed as

$$\mathbf{C}_{id} = \mathbf{Q}_i \begin{bmatrix} \mathbf{C}_{ss} \\ \mathbf{C}_{dd} \end{bmatrix}. \quad (\text{B.3})$$

By substituting (B.2) and (B.3) for (B.1), we obtain

$$\text{Var}(\mathbf{m}_i|\mathbf{M})/\sigma_v^2 = \mathbf{C}_{ii} - \mathbf{Q}_i \begin{bmatrix} \mathbf{C}_{ss} & \mathbf{C}_{sd} \\ \mathbf{C}_{ds} & \mathbf{C}_{dd} \end{bmatrix} \mathbf{Q}_i'.$$

APPENDIX C

Proof that \mathbf{D} is block diagonal

From (5) in the text, we have

$$\text{Cov}(\mathbf{v}_i, \mathbf{v}_j|\mathbf{M})/\sigma_v^2 = \mathbf{C}_{ij} = \mathbf{Q}_i \begin{bmatrix} \mathbf{C}_{sij} \\ \mathbf{C}_{dij} \end{bmatrix} \quad (\text{C.1})$$

and from (10), we have

$$\mathbf{v}_i = \mathbf{Q}_i \begin{bmatrix} \mathbf{v}_{s_i} \\ \mathbf{v}_{d_i} \end{bmatrix} + \mathbf{m}_i \quad \text{and} \quad \mathbf{v}_j = \mathbf{Q}_j \begin{bmatrix} \mathbf{v}_{s_j} \\ \mathbf{v}_{d_j} \end{bmatrix} + \mathbf{m}_j \quad (\text{C.2})$$

where s_i and d_i are the sire and dam of individual i , respectively. To prove that \mathbf{D} is block diagonal, it is sufficient to show that the covariance between \mathbf{m}_i and \mathbf{m}_j is null.

From (C.2),

$$\begin{aligned} \sigma_v^2 \mathbf{C}_{ij} &= \text{Cov} \left(\mathbf{Q}_i \begin{bmatrix} \mathbf{v}_{s_i} \\ \mathbf{v}_{d_i} \end{bmatrix} + \mathbf{m}_i, \mathbf{v}_j' | \mathbf{M} \right) \\ &= \mathbf{Q}_i \text{Cov} \left(\begin{bmatrix} \mathbf{v}_{s_i} \\ \mathbf{v}_{d_i} \end{bmatrix}, \mathbf{v}_j' | \mathbf{M} \right) + \text{Cov}(\mathbf{m}_i, \mathbf{v}_j' | \mathbf{M}) \\ &= \mathbf{Q}_i \begin{bmatrix} \mathbf{C}_{sij} \\ \mathbf{C}_{dij} \end{bmatrix} + \text{Cov}(\mathbf{m}_i, \mathbf{v}_j' | \mathbf{M}). \end{aligned} \quad (\text{C.3})$$

But from equation (C.1), we have $\sigma_v^2 \mathbf{C}_{ij} = \mathbf{Q}_i \begin{bmatrix} \mathbf{C}_{sij} \\ \mathbf{C}_{dij} \end{bmatrix}$, which leads us to conclude that the second term in (C.3), $\text{Cov}(\mathbf{m}_i, \mathbf{v}'_j | \mathbf{M})$, must be zero. Similarly, $\text{Cov}(\mathbf{m}_i, \mathbf{v}'_{s_j} | \mathbf{M})$ and $\text{Cov}(\mathbf{m}_i, \mathbf{v}'_{d_j} | \mathbf{M})$ could be shown to be null. Finally, given that $\text{Cov}(\mathbf{m}_i, \mathbf{v}'_j | \mathbf{M})$, $\text{Cov}(\mathbf{m}_i, \mathbf{v}'_{s_j} | \mathbf{M})$, and $\text{Cov}(\mathbf{m}_i, \mathbf{v}'_{d_j} | \mathbf{M})$ are all null, $\text{Cov}(\mathbf{v}_i, \mathbf{v}'_j | \mathbf{M})$ must be null.