BioOne COMPLETE

Updating genetic relationship matrices and their inverses: a methodology note

Author: Nilforooshan, Mohammad Ali

Source: Canadian Journal of Animal Science, 100(2) : 292-298

Published By: Canadian Science Publishing

URL: https://doi.org/10.1139/cjas-2019-0106

BioOne Complete (complete.BioOne.org) is a full-text database of 200 subscribed and open-access titles in the biological, ecological, and environmental sciences published by nonprofit societies, associations, museums, institutions, and presses.

Your use of this PDF, the BioOne Complete website, and all posted and associated content indicates your acceptance of BioOne's Terms of Use, available at www.bioone.org/terms-of-use.

Usage of BioOne Complete content is strictly limited to personal, educational, and non - commercial use. Commercial inquiries or rights and permissions requests should be directed to the individual publisher as copyright holder.

BioOne sees sustainable scholarly publishing as an inherently collaborative enterprise connecting authors, nonprofit publishers, academic institutions, research libraries, and research funders in the common goal of maximizing access to critical research.

Downloaded From: https://bioone.org/journals/Canadian-Journal-of-Animal-Science on 28 Mar 2024
Terms of Use: https://bioone.org/terms-of-use

Pagination not final (cite DOI) / Pagination provisoire (citer le DOI)

NRC Research Press

ARTICLE

Updating genetic relationship matrices and their inverses: a methodology note

Mohammad Ali Nilforooshan

Abstract: Most of the calculations for genetic relationship matrices and their inverses utilized in recent ($t+1$) evaluations for populations of interest can be avoided by updating these matrices at t evaluations for only new animals. This study describes and develops existing methods for updating pedigree-based and marker-based relationship matrices and their inverses. Updating some of the matrices could benefit from parallel computing.

Key words: relationship matrix, inverse, updating, parallel processing.

Résumé : La plupart des calculs de matrices des relations génétiques et leurs inverses utilisés dans les évaluations les plus actuelles ($t+1$) pour les populations étudiées peuvent être évités en mettant à jour ces matrices à des évaluations t pour seulement les nouveaux animaux. Cette étude décrit et développe les méthodes existantes pour la mise à jour des matrices de relations à base de pedigree et de marqueurs et leurs inverses. La mise à jour de certaines de ces matrices pourrait bénéficier du traitement parallèle. [Traduit par la Rédaction]

Mots-clés : matrice des relations, inverse, mise à jour, traitement parallèle.

Introduction

Best linear unbiased prediction (BLUP; Henderson 1973), genomic BLUP (GBLUP; VanRaden 2008), and single-step GBLUP (ssGBLUP; Aguilar et al. 2010; Christensen and Lund 2010) are the most common methods used for genetic and (or) genomic evaluation of livestock. Relationships among individuals are modelled via pedigree or genomic relationship matrices (**A** or **G**, respectively). The inverse of these matrices are required in BLUP and GBLUP, respectively. In ssGBLUP, both inverses together with the inverse of the pedigree relationship matrix for genotyped animals are required. A major computational burden for genetic evaluation systems is matrix inversion. The inversion of **G** has a cubic computational cost proportional to the number of genotyped animals (Meyer et al. 2013), using the conventional matrix inversion algorithms. However, the method for indirect inversion of **A** (Henderson 1975; Quaas 1976) has made its inversion possible at a linear cost. There is an opportunity to reduce the computation costs at time $t+1$ evaluations by updating the relationship matrices and their inverses at time t for the information on the new animals. In this methodology note, methods of updating relationship matrices and their inverses are presented.

Methods

Updating G

The most common way of forming the genomic relationship matrix is $\mathbf{G} = c\mathbf{Z}\mathbf{Z}'$ (VanRaden 2008), where $c = 1/[2\Sigma p_i(1-p_i)]$ for scaling **G** to be analogous to **A**, p_i is the allele frequency at locus i, $\mathbf{Z} = \mathbf{M} - 2\mathbf{P}$, where **M** is the genotype matrix with genotypes at each loci coded as {0, 1 (heterozygote), 2}, $\mathbf{P} = \mathbf{1}\boldsymbol{p}'$, where $\boldsymbol{p}$ is a column vector of p_i (from the previous evaluation), and $\mathbf{1}$ is a column vector of ones with the order of the number of animals. Denoting old and young animals with 1 and 2, the genomic relationship matrix can be updated for new genotypes:

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{pmatrix} = c\begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix}\begin{pmatrix} \mathbf{Z}_1' & \mathbf{Z}_2' \end{pmatrix} = c\begin{pmatrix} \mathbf{Z}_1\mathbf{Z}_1' & \mathbf{Z}_1\mathbf{Z}_2' \\ \mathbf{Z}_2\mathbf{Z}_1' & \mathbf{Z}_2\mathbf{Z}_2' \end{pmatrix} \quad (1)$$

Received 17 June 2019. Accepted 22 November 2019.

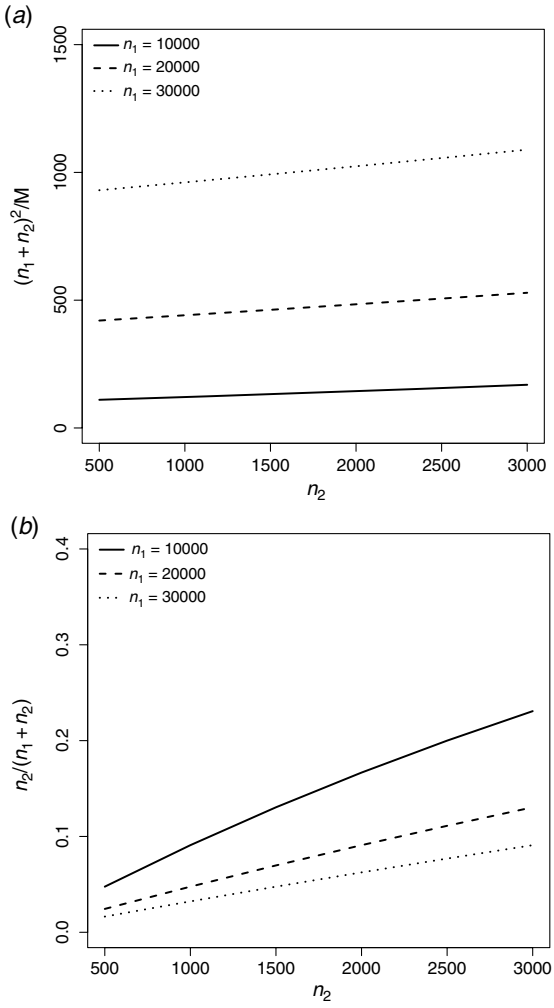
M.A. Nilforooshan. Livestock Improvement Corporation, Private Bag 3016, Hamilton 3240, New Zealand.

Email for correspondence: mohammad.nilforooshan@lic.co.nz.

Copyright remains with the author(s) or their institution(s). This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

Downloaded From: https://bioone.org/journals/Canadian-Journal-of-Animal-Science on 28 Mar 2024
Terms of Use: https://bioone.org/terms-of-use

Fig. 1. The impact of various n_1 and n_2 on (*a*) $(n_1+n_2)^2$ in millions (M) and (*b*) $n_2/(n_1+n_2)$.



Matrices $\mathbf{G}_{12}$ and $\mathbf{G}_{22}$ are required for updating $\mathbf{G}$:

$$(2)\quad \begin{pmatrix}\mathbf{G}_{12}\\ \mathbf{G}_{22}\end{pmatrix} = c\begin{pmatrix}\mathbf{Z}_1\mathbf{Z}_2'\\ \mathbf{Z}_2\mathbf{Z}_2'\end{pmatrix} = c\begin{pmatrix}\mathbf{Z}_1\\ \mathbf{Z}_2\end{pmatrix}\mathbf{Z}_2' = c\mathbf{Z}\mathbf{Z}_2' = c(\mathbf{M}-2\mathbf{P})(\mathbf{M}_2-2\mathbf{P}_2)'$$

where $\mathbf{M}_2$ and $\mathbf{P}_2$ are the rows of $\mathbf{M}$ and $\mathbf{P}$ for young animals. In comparison with $\mathbf{Z}\mathbf{Z}'$, $\mathbf{Z}\mathbf{Z}_2'$ has $(n_1+n_2)mn_2$ computational complexity (O) rather than $(n_1+n_2)^2m$, where n_1, n_2, and m are the number of old genotyped animals, young genotyped animals, and markers, respectively. One unit of O is defined as a single arithmetic operation. Figure 1 shows (*a*) $O(\mathbf{Z}\mathbf{Z}')$ for $m=1$ and (*b*) $O(\mathbf{Z}\mathbf{Z}_2')/O(\mathbf{Z}\mathbf{Z}') = n_2/(n_1+n_2)$ for various n_1 and n_2. For $n_2 < n_1$, $O(\mathbf{Z}\mathbf{Z}')$ changed linearly by n_2 and exponentially increased by increasing n_1 (Fig. 1*a*). The benefit from updating $\mathbf{G}$ increased by increasing n_1/n_2 (Fig. 1*b*).

Considering $\mathbf{P}' = [\mathbf{P}_1'\ \mathbf{P}_2']$, if the old and new genotypes are from discrete populations or breeds, allele frequencies in $\mathbf{P}_1$ (rows of $\mathbf{P}$ for old genotypes) and in $\mathbf{P}_2$ are different (i.e., $\mathbf{P}_1 = \mathbf{1}_{n_1}\boldsymbol{p}$ and $\mathbf{P}_2 = \mathbf{1}_{n_2}\boldsymbol{p}_2$). In that case,

$$(3)\quad \begin{pmatrix}\mathbf{G}_{12}\\ \mathbf{G}_{22}\end{pmatrix} = \begin{bmatrix}\sqrt{c}(\mathbf{M}_1-2\mathbf{P}_1)\\ \sqrt{c_2}(\mathbf{M}_2-2\mathbf{P}_2)\end{bmatrix}\sqrt{c_2}(\mathbf{M}_2-2\mathbf{P}_2)'$$

Even for homogeneous populations, $\boldsymbol{p}$ and $\boldsymbol{p}_2$ can be different. If n_2 is relatively large and deviations between $\boldsymbol{p}$ and $\boldsymbol{p}_2$ are considerable, $\boldsymbol{p}$ should be updated to $\boldsymbol{p}^* = (n_1\boldsymbol{p}+n_2\boldsymbol{p}_2)/(n_1+n_2)$, and c should be updated to c^* (using $\boldsymbol{p}^*$) for both groups of animals. That updates eq. 2 and results in eq. 4:

$$(4)\quad \begin{pmatrix}\mathbf{G}_{12}\\ \mathbf{G}_{22}\end{pmatrix} = c^*(\mathbf{M}-2\mathbf{P}^*)(\mathbf{M}_2-2\mathbf{P}_2^*)'$$

Furthermore, allele frequencies in $\mathbf{G}_{11}$ should be updated, which is done by regressing $\mathbf{G}_{11}$ as if $\boldsymbol{p}^*$ was applied ($\mathbf{G}_{11}^*$) to $\mathbf{G}_{11}$ (i.e., $\mathbf{G}_{11}^* = b\mathbf{G}_{11} + \mathbf{1}\mathbf{1}'\alpha$), where $b = c^*/c$ and $\alpha = 4c^*[\boldsymbol{p}^{*\prime}\boldsymbol{p}^* - \boldsymbol{p}'\boldsymbol{p} - (\boldsymbol{p}^*-\boldsymbol{p})'\mathbf{M}_1'\mathbf{1}/n_1]$. The proof of α is provided in the Appendix A. Like any regression, error terms are involved (in $\mathbf{G}_{11}^*$). Therefore, $\mathbf{G}_{11}^*$ is an approximation of $\mathbf{G}_{11}$ as if $\boldsymbol{p}^*$ was used instead of $\boldsymbol{p}$. Vitezica et al. (2011) used a similar approach to regress $\mathbf{G}$ to $\mathbf{A}_{gg}$ (block of $\mathbf{A}$ corresponding to genotyped animals) in the context of ssGBLUP to regress $\mathbf{G}$ to the same base population as in $\mathbf{A}_{gg}$. Given $\boldsymbol{p}$ and $\boldsymbol{p}^*$, $O(\alpha) = (n_1+4)(m+1)-2$, from which c^*, $\boldsymbol{p}^{*\prime}\boldsymbol{p}^*$, and $\boldsymbol{p}'\boldsymbol{p}$ have computational complexities of m and $O[(\boldsymbol{p}^*-\boldsymbol{p})'\mathbf{M}_1'] = m+mn_1$. Figure 2 shows the impact of different m and n_1 on $O(\alpha)$.

Updating $\mathbf{G}^{-1}$

Hager (1989) introduced the state-of-the-art method of updating the inverse of a matrix. Meyer et al. (2013) showed how this method can be used to update $\mathbf{G}^{-1}$, which is presented below.

$$(5)\quad \begin{aligned}\mathbf{G}^{-1} &= \begin{pmatrix}\mathbf{G}_{11} & \mathbf{G}_{12}\\ \mathbf{G}_{21} & \mathbf{G}_{22}\end{pmatrix}^{-1}\\ &= \begin{pmatrix}\mathbf{G}_{11}^{-1}+\mathbf{Q}'\mathbf{G}^{22}\mathbf{Q} & -\mathbf{Q}'\mathbf{G}^{22}\\ -\mathbf{G}^{22}\mathbf{Q} & \mathbf{G}^{22}\end{pmatrix}\end{aligned}$$

where $\mathbf{G}^{22} = (\mathbf{G}_{22}-\mathbf{Q}\mathbf{G}_{12})^{-1}$ and $\mathbf{Q} = \mathbf{G}_{21}\mathbf{G}_{11}^{-1}$. There are other forms of presenting the updated $\mathbf{G}^{-1}$, such as

$$(6)\quad \mathbf{G}^{-1} = \begin{pmatrix}\mathbf{G}_{11}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix} + \begin{pmatrix}-\mathbf{Q}'\\ \mathbf{I}\end{pmatrix}\mathbf{G}^{22}\begin{pmatrix}-\mathbf{Q} & \mathbf{I}\end{pmatrix}$$

and

$$(7)\quad \mathbf{G}^{-1} = \begin{pmatrix}\mathbf{I} & -\mathbf{Q}'\\ \mathbf{0} & \mathbf{I}\end{pmatrix}\begin{pmatrix}\mathbf{G}_{11}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{G}^{22}\end{pmatrix}\begin{pmatrix}\mathbf{I} & \mathbf{0}\\ -\mathbf{Q} & \mathbf{I}\end{pmatrix}$$

where $\mathbf{I}$ is an identity matrix with the order of the number of new genotyped animals (n_2). The only matrix in need of inversion is $\mathbf{G}_{22}-\mathbf{Q}\mathbf{G}_{12}$, which is a small matrix because usually the number of additional genotypes is small. This method considers no changes in allele frequencies due to new genotypes. Assuming allele frequencies in the new genotyped animals causing change in the scale, but not the base of $\mathbf{G}$, this method

Downloaded From: https://bioone.org/journals/Canadian-Journal-of-Animal-Science on 28 Mar 2024
Terms of Use: https://bioone.org/terms-of-use

Pagination not final (cite DOI) / Pagination provisoire (citer le DOI)

Fig. 2. The impact of various n_1 and m on $(n_1+4)(m+1)-2$ in millions (M).

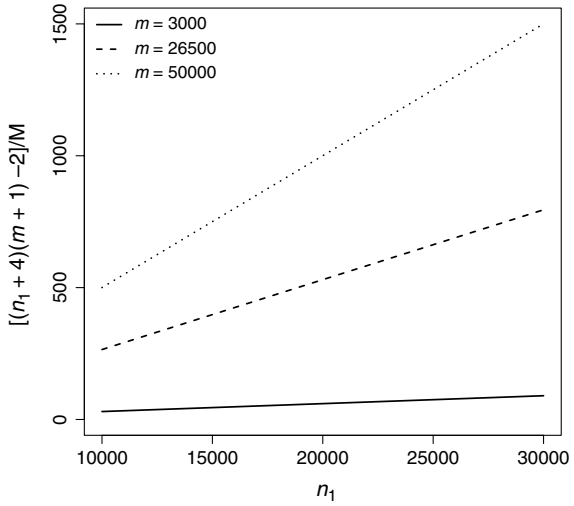
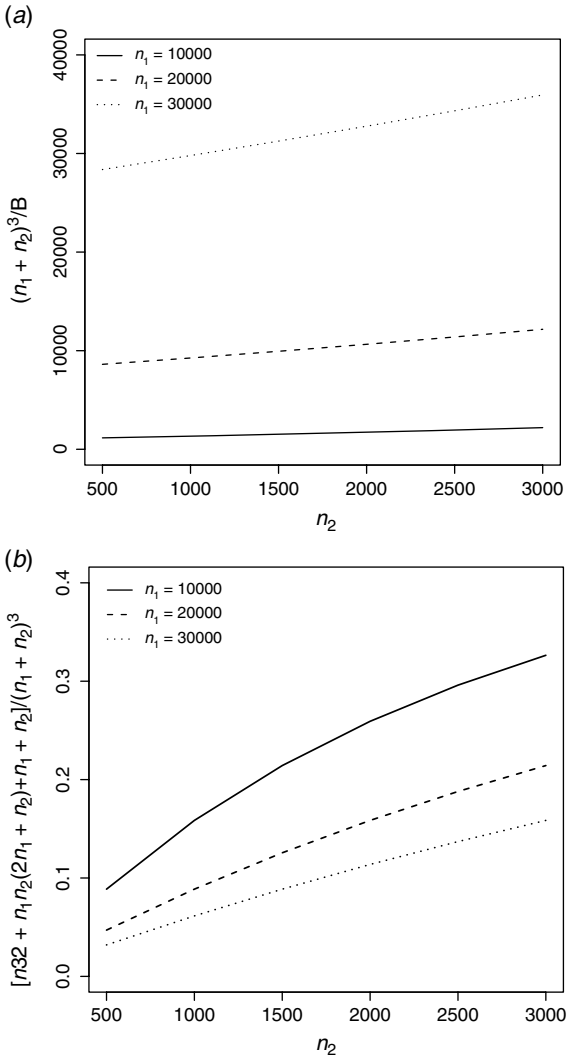


Fig. 3. The impact of various n_1 and n_2 on (*a*) $(n_1+n_2)^3$ in billions (B) and (*b*) $[n_2^3+n_1n_2(2n_1+n_2)+n_1+n_2]/(n_1+n_2)^3$.



can be modified by obtaining $\mathbf{G}_{12}$ and $\mathbf{G}_{22}$ according to eq. 4, and multiplying $\mathbf{G}_{11}^{-1}$ by c/c^*.

The equation for updating $\mathbf{G}^{-1}$ (eq. 5) is similar to the equation for the inverse of the pedigree relationship matrix including phantom parent groups, introduced by Quaas (1988). By substituting $\mathbf{G}^{22}$ with $\mathbf{A}^{-1}$, $\mathbf{G}_{11}^{-1}$ with $\mathbf{\Phi}_{22}$, and redefining $\mathbf{Q}$ as the relationship coefficient matrix between the base animals and their phantom parent groups, eq. 5 is then changed to

$$(8)\qquad \begin{pmatrix} \mathbf{\Phi}_{22}+\mathbf{Q}'\mathbf{A}^{-1}\mathbf{Q} & -\mathbf{Q}'\mathbf{A}^{-1} \\ -\mathbf{A}^{-1}\mathbf{Q} & \mathbf{A}^{-1} \end{pmatrix}$$

Considering $\mathbf{\Phi}_{22}=\mathbf{0}$, this matrix becomes the same as the inverse of the pedigree relationship matrix including phantom parent groups (Quaas 1988). This assumption is true because there is no previous inverse for new animals (phantom parent groups) to be added.

Compared with inverting the whole $\mathbf{G}$, updating $\mathbf{G}^{-1}$ reduces the matrix inversion complexity from $(n_1+n_2)^3$ to n_2^3. However, there are additional complexities for matrix multiplication, equal to $n_1^2n_2$ for $\mathbf{Q}$, $n_1n_2^2$ for $\mathbf{G}^{22}\mathbf{Q}$, $n_1^2n_2$ for $\mathbf{Q}'\mathbf{G}^{22}\mathbf{Q}$, and for matrix summations equal to n_1+n_2. The greater the n_1, and the smaller the n_2, the more the advantage in updating $\mathbf{G}^{-1}$ (Meyer et al. 2013). Figure 3 shows (*a*) $O(\mathbf{G}^{-1})$ and (*b*) O(updating $\mathbf{G}_{11}^{-1}$ to $\mathbf{G}^{-1}$)/$O(\mathbf{G}^{-1})=[n_2^3+n_1n_2(2n_1+n_2)+n_1+n_2]/(n_1+n_2)^3$ with various n_1 and n_2. Generally, computational complexity of $\mathbf{G}^{-1}$ increases exponentially by increasing n_1 and n_2. However, with n_2 smaller than n_1, the trend of $O(\mathbf{G}^{-1})$ by n_2 gets closer to linear (Fig. 3*a*). Larger n_1 also caused $O(\mathbf{G}^{-1})$ to increase at a higher rate by increasing n_2. The benefit from updating $\mathbf{G}^{-1}$ increased by increasing n_1/n_2 (Fig. 3*b*). The strategy of updating $\mathbf{G}^{-1}$ can be applied for updating $\mathbf{A}_{gg}^{-1}$ used in ssGBLUP (Aguilar et al. 2010; Christensen and Lund 2010). However, it would be more convenient to update $\mathbf{A}^{gg}-\mathbf{A}_{gg}^{-1}$, where $\mathbf{A}^{gg}$ is the block of $\mathbf{A}^{-1}$ for genotyped animals. The matrix $\mathbf{G}^{-1}+\mathbf{A}^{gg}-\mathbf{A}_{gg}^{-1}$ is used in ssGBLUP instead of $\mathbf{A}^{gg}$ in BLUP. According to Strandén and Mäntysaari (2014):

$$(9)\qquad \mathbf{A}^{gg}-\mathbf{A}_{gg}^{-1}=\mathbf{A}^{gn}(\mathbf{A}^{nn})^{-1}\mathbf{A}^{ng}$$

where n denotes non-genotyped animals. The $\mathbf{A}^{ng}$ and $\mathbf{A}^{nn}$ blocks are easy to obtain. Instead of directly inverting $\mathbf{A}^{nn}$, each column of $(\mathbf{A}^{nn})^{-1}\mathbf{A}^{ng}$ can be obtained by solving the equation system $\mathbf{A}^{nn}\boldsymbol{x}=\boldsymbol{y}$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the jth columns of $(\mathbf{A}^{nn})^{-1}\mathbf{A}^{ng}$ and $\mathbf{A}^{ng}$, respectively. Therefore, to get an updated $\mathbf{A}^{gg}-\mathbf{A}_{gg}^{-1}$, the new genotyped animals are appended to $\mathbf{A}^{ng}$ in the above formula. However, this is assuming that the pedigree of the new genotyped animals does not contain any new non-genotyped animal.

A better approach for updating $\mathbf{G}^{-1}$ is using the algorithm for proven and young (APY), developed by Misztal et al. (2014). This algorithm is better in the sense that it is computationally more feasible, it has less noise

Downloaded From: https://bioone.org/journals/Canadian-Journal-of-Animal-Science on 28 Mar 2024
Terms of Use: https://bioone.org/terms-of-use

Pagination not final (cite DOI) / Pagination provisoire (citer le DOI)

associated with noncoding markers (Nilforooshan and Lee 2019), and it may overcome the singularity problem of **G**, especially when the number of genotyped animals reach the number of markers. The $\mathbf{G}^{-1}_{APY}$ approximate of $\mathbf{G}^{-1}$ is calculated as (Misztal et al. 2014)

$$(10)\quad \mathbf{G}^{-1}_{APY} = \begin{bmatrix} \mathbf{G}^{-1}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} -\mathbf{G}^{-1}_{11}\mathbf{G}_{12} \\ \mathbf{I} \end{bmatrix} \mathbf{D}^{-1}_{22} \begin{bmatrix} -\mathbf{G}_{21}\mathbf{G}^{-1}_{11} & \mathbf{I} \end{bmatrix}$$

where $\mathbf{D}_{22}$ is a diagonal matrix with diagonal elements $D_{ii} = g_{ii} - \mathbf{g}_{i1}\mathbf{G}^{-1}_{11}\mathbf{g}_{1i}$, where g_{ii} is the ith diagonal element of $\mathbf{G}_{22}$, n is the number of markers, m_{ij} is the genotype of the new individual i for marker j, and $\mathbf{g}_{i1}$ is the ith row of $\mathbf{G}_{21}$. Thus, given $\mathbf{G}^{-1}_{11}$ from the previous evaluation, $\mathbf{G}_{12}$ and g_{ii} elements are required, where $\mathbf{G}_{12} = c(\mathbf{M}_1 - 2\mathbf{P}_1)(\mathbf{M}_2 - 2\mathbf{P}_2)'$, and $g_{ii} = c\sum_{j=1}^{n}(m_{ij} - 2p_j)^2$. Please note that forming and inverting $\mathbf{G}_{22}$ are not required.

In APY, genotyped animals are divided into core and noncore animals and the direct inversion is only required for the block of **G** for core animals (Misztal et al. 2014). This method can overcome the computational challenges of inverting a large **G**. In addition, the closer the number of genotyped animals get to the number of markers, the numerical stability of **G** decreases until the number of genotyped animals exceed the number of markers. At this point, **G** would be singular. By updating $\mathbf{G}^{-1}$ using APY, previously genotyped animals are considered as core and the new genotyped animals are considered as noncore. If genotypes from the core animals provide enough information about the independent chromosome segments in the population, the core information is provided and the off-diagonals of $\mathbf{G}^{22}$ do not contribute to the accuracy and can be ignored. Thus, the genomic breeding value of noncore animals is conditioned to the genomic breeding value of core animals (Misztal 2016). Random cross-generation core definition has been found to perform well in APY (Ostersen et al. 2016; Bradford et al. 2017; Nilforooshan and Lee 2019). Updating $\mathbf{G}^{-1}$ consecutively with APY results in dropping the latest generations from the core sample, which may have unfavourable effects. Therefore, after one or a few (depending on the population) APY updates of $\mathbf{G}^{-1}$, it is recommended to replace $\mathbf{G}^{-1}_{11}$ with a new $\mathbf{G}^{-1}$ or with a random core $\mathbf{G}^{-1}_{APY}$.

Parallel processing

Computational time for matrix operations can be considerably reduced by parallel processing. Computational complexity of a $n_1 \times m$ by $m \times n_2$ matrix multiplication is $n_1 m n_2$. The $n_1 m n_2$ computational complexity can split across n_1, n_2, or n_3 parallel processes, where n_3 is an integer less than n_1 and n_2. Matrix multiplication can be done in independent parallel processes; thus, the cost of communications across nodes is minimized. For example, row i of **AB** can be obtained independently from other rows of **AB** by multiplying row i of **A** to **B**, or column j of **AB** can be obtained independently from other columns of **AB** by multiplying **A** to column j of **B**. An R function for parallel processing of matrix multiplication is provided in Appendix B. Though not discussed here, parallel processing algorithms do exist for matrix inversion and are used in some genetic evaluation softwares.

Updating A

Matrix **A** is not needed in genetic evaluation models and its calculation is computationally more difficult than calculating $\mathbf{A}^{-1}$. However, it is usually needed for postevaluation procedures, such as designing breeding schemes, preserving genetic variation, and controlling inbreeding in the population. Updating **A** is not different from resuming an incomplete construction of **A** to its full matrix. The following algorithm demonstrates a procedure for updating **A**. Considering no pedigree errors (e.g., young animals being parents to old animals) and no pedigree correction, the pedigree file for young animals has only young animals in the first (animal ID) column. Any row corresponding to animals not among the young animals is deleted. Then, **A** is updated with the following simple rules:

1. For animals in the new pedigree with no parents:
   1.1. Append elements of 1 to **A** for the corresponding diagonal values.
   1.2. Delete those animals from the new pedigree.
2. While the new pedigree is not empty:
   2.1. Find an animal with parent(s) not available in the first column of the new pedigree.
   2.2. Following Emik and Terrill (1949), the relationship of animal i with others (already) in **A** is the average of the relationship of its parents, s and d with those animals. The same is true for the relationship of the animal to its parents [$A_{is} = (A_{ss} + A_{sd})/2$, $A_{id} = (A_{dd} + A_{sd})/2$], and $A_{ii} = 1 + A_{sd}/2$.
   2.3. Delete that animal from the new pedigree.

This technique can be used, not only for updating, but also for constructing **A** from scratch. Sorting animals by age is not needed as it is built into the algorithm by picking animals in the right order (parents before progeny). Instead of updating **A**, which may include millions of animals, a subset of it including the animals of interest (e.g., the last x generations) and the parents of the new animals (if known) can be updated.

There are available methods for calculating inbreeding coefficients (Tier 1990; Meuwissen and Luo 1992; Colleau 2002; Sargolzaei and Iwaisaki 2004; R.L. Quaas (1995), unpublished note) that can be adopted in updating situations (i.e., calculating inbreeding coefficients for young animals given inbreeding coefficients for old animals). Sargolzaei and Iwaisaki (2005) compared computational performance of these four algorithms. The algorithms of R.L. Quaas (1995, unpublished note) and Sargolzaei and Iwaisaki (2004) are modifications to the algorithm of Meuwissen and Luo (1992). The performance of the algorithms depended on the number of generations, population, and family sizes. The algorithm of Sargolzaei and

Downloaded From: https://bioone.org/journals/Canadian-Journal-of-Animal-Science on 28 Mar 2024
Terms of Use: https://bioone.org/terms-of-use

Pagination not final (cite DOI) / Pagination provisoire (citer le DOI)

Iwaisaki (2004) was faster than the other algorithms. Computation time considerably decreased in updating situations, in which the algorithm of Sargolzaei and Iwaisaki (2004) outperformed the other algorithms (Sargolzaei and Iwaisaki 2005). Colleau (2002) developed an indirect method for obtaining individual inbreeding coefficients and relationship statistics in the population. The method was indirect because instead of element-wise calculation of the relationship matrix, groups of elements were calculated simultaneously. The computational efficiency of the method was heavily reliant on the sparseness of the inverse of the relationship matrix.

Computational complexity of calculating or updating A depends on many factors and it differs from population to population and animal to animal. The computation involves finding the parents of the animal; searching for their relationships, the size, and the sparsity of **A**; the number of previous animals in the pedigree; the number of new animals; and the computational algorithm.

Updating $\mathbf{A}^{-1}$

Updating $\mathbf{A}^{-1}$ is not different from resuming an incomplete construction of $\mathbf{A}^{-1}$ to its full matrix. However, computationally, reading and updating an old $\mathbf{A}^{-1}$ is not justifiable over calculating $\mathbf{A}^{-1}$ from scratch. Calculation of $\mathbf{A}^{-1}$ is easier than the calculation of **A**. The elements of $\mathbf{A}^{-1}$ for an animal are conditional to its parents, whereas the elements of **A** for an animal are also conditional to the other relatives.

Matrix storage

Conventionally, relationship matrices and their inverses are saved in a sparse upper- and (or) lower-triangle format with three columns for the ID of animals and the matrix element. It saves disk space and makes indexing matrix elements easy. However, symmetric dense matrices can benefit from being stored as an array of upper- and (or) lower-triangular values, which takes considerably less storage than a three-column data frame. In programming, this method of storing a matrix is called packed storage. This method is used in Fortran's linear algebra libraries, BLAS, and LAPACK (Wikipedia Contributors 2013). For a packed matrix of length n, the matrix dimension is obtained as $N = [\sqrt{(1+8n)} - 1]/2$. For indexing, each row i starts with the $[(i-1)(N - i/2 + 1) + 1]$th element and ends with the $[(2N - i + 1)i/2]$th element of the vector, and an element from the ith row and jth column of the matrix is located at the $[(i-1)(N - i/2 + 1) + 1 + |i - j|]$th element of the vector.

Updating vs. rebuilding

On deciding whether to update matrices or computing them from scratch, some consideration should be given to the following:

1. The drive's read speed should be considered. Nowadays, there are commercial non-volatile memory express drives with a read speed over 5000 MB per second. There is less concern about the writing speed as it occurs after computations have been completed. In efficient programming, read-write instances are minimized.
2. The size and the sparsity of the matrix to be read compared with the size of the previous data file (e.g., pedigree or genotypes) to rebuild the matrix.
3. Data file format (e.g., binary vs. ASCII).
4. Data reading strategy (i.e., mapping the file into memory vs. reading the file into a buffer via a connection; reading a large file by memory mapping is faster). The pros and cons of specific data reading strategies are not in the scope of this paper.
5. Computational complexity of the calculations.
6. Central processing unit clock speed.
7. Single-core vs. parallel computing.
8. Storage vs. computation cost.

Conclusion

An overview of the methods for updating different relationship matrices and their inverses was provided and possible improvements were proposed. There are possibilities for reducing required computational time and resources for calculating relationship matrices or their inverses by updating the previously calculated matrix for new animals and parallel computing. The downside of updating matrices is allocating disk space for saving the matrix at time t and reading it for the evaluation at time $t+1$. Both can be reduced by saving the old matrix as half-stored (i.e., upper and (or) lower diagonal), sparse (i.e., skipping zero elements), and binary.

Acknowledgements

The author acknowledges Livestock Improvement Corporation (LIC, Hamilton, NZ) for providing funds for the open access publication of this paper.

References

Aguilar, I., Misztal, I., Johnson, D.L., Legarra, A., Tsuruta, S., and Lawlor, T.J. 2010. Hot topic: a unified approach to utilize phenotypic, full pedigree, and genomic information for genetic evaluation of Holstein final score. J. Dairy Sci. **93**: 743–752. doi:10.3168/jds.2009-2730. PMID:20105546.

Bradford, H.L., Pocrnić, I., Fragomeni, B.O., Lourenco, D.A.L., and Misztal, I. 2017. Selection of core animals in the algorithm for proven and young using a simulation model. J. Anim. Breed. Genet. **134**: 545–552. doi:10.1111/jbg.12276. PMID:28464315.

Christensen, O.F., and Lund, M.S. 2010. Genomic prediction when some animals are not genotyped. Genet. Sel. Evol. **42**: 2. doi:10.1186/1297-9686-42-2. PMID:20105297.

Colleau, J.-J. 2002. An indirect approach to the extensive calculation of relationship coefficients. Genet. Sel. Evol. **34**: 409. doi:10.1186/1297-9686-34-4-409. PMID:12270102.

Emik, L.O., and Terrill, C.E. 1949. Systematic procedures for calculating inbreeding coefficients. J. Hered. **40**: 51–55. doi:10.1093/oxfordjournals.jhered.a105986. PMID:18116093.

Hager, W.W. 1989. Updating the inverse of a matrix. SIAM Rev. **31**: 221–239. doi:10.1137/1031049.

Henderson, C.R. 1973. Sire evaluation and genetic trends. J. Anim. Sci. 1973: 10–41. doi:10.1093/ansci/1973.Symposium.10.

Downloaded From: https://bioone.org/journals/Canadian-Journal-of-Animal-Science on 28 Mar 2024
Terms of Use: https://bioone.org/terms-of-use

Pagination not final (cite DOI) / Pagination provisoire (citer le DOI)

Henderson, C.R. 1975. Rapid method for computing the inverse of a relationship matrix. J. Dairy Sci. **58**: 1727–1730. doi:10.3168/jds.S0022-0302(75)84776-X.

Meuwissen, T.H.E., and Luo, Z. 1992. Computing inbreeding coefficients in large populations. Genet. Sel. Evol. **24**: 305–313. doi:10.1186/1297-9686-24-4-305.

Meyer, K., Tier, B., and Graser, H.U. 2013. Technical note: updating the inverse of the genomic relationship matrix. J. Anim. Sci. **91**: 2583–2586. doi:10.2527/jas.2012-6056. PMID:23508030.

Misztal, I. 2016. Inexpensive computation of the inverse of the genomic relationship matrix in populations with small effective population size. Genetics, **202**: 401–409. doi:10.1534/genetics.115.182089. PMID:26584903.

Misztal, I., Legarra, A., and Aguilar, I. 2014. Using recursion to compute the inverse of the genomic relationship matrix. J. Dairy Sci. **97**: 3943–3952. doi:10.3168/jds.2013-7752. PMID:24679933.

Nilforooshan, M.A., and Lee, M. 2019. The quality of the algorithm for proven and young with various sets of core animals in a multibreed sheep population. J. Anim. Sci. **97**: 1090–1100. doi:10.1093/jas/skz010. PMID:30624671.

Ostersen, T., Christensen, O.F., Madsen, P., and Henryon, M. 2016. Sparse single-step method for genomic evaluation in pigs. Genet. Sel. Evol. **48**: 48. doi:10.1186/s12711-016-0227-8. PMID:27357825.

Quaas, R.L. 1976. Computing the diagonal elements and inverse of a large numerator relationship matrix. Biometrics, **32**: 949–953. doi:10.2307/2529279.

Quaas, R.L. 1988. Additive genetic model with groups and relationships. J. Dairy Sci. **71**: 1338–1345. doi:10.3168/jds.S0022-0302(88)79691-5.

Sargolzaei, M., and Iwaisaki, H. 2004. An efficient algorithm for computing inbreeding coefficients in large populations. Jpn. J. Biom. **25**: 25–36. doi:10.5691/jjb.25.25.

Sargolzaei, M., and Iwaisaki, H. 2005. Comparison of four direct algorithms for computing inbreeding coefficients. Anim. Sci. J. **76**: 401–406. doi:10.1111/j.1740-0929.2005.00282.x.

Strandén, I., and Mäntysaari, E.A. 2014. Comparison of some equivalent equations to solve single-step GBLUP. Page, 69 in Proc. 10th World Congress of Genetics Applied to Livestock Production, 17–22 Aug. 2014, Vancouver, BC, Canada.

Tier, B. 1990. Computing inbreeding coefficients quickly. Genet. Sel. Evol. **22**: 419–430. doi:10.1186/1297-9686-22-4-419.

VanRaden, P.M. 2008. Efficient methods to compute genomic predictions. J. Dairy Sci. **91**: 4414–4423. doi:10.3168/jds.2007-0980. PMID:18946147.

Vitezica, Z.G., Aguilar, I., Misztal, I., and Legarra, A. 2011. Bias in genomic predictions for populations under selection. Genet. Res. **93**: 357–366. doi:10.1017/S001667231100022X. PMID:21767459.

Wikipedia Contributors. 2013. Packed storage matrix. [Online]. Available from https://en.wikipedia.org/w/index.php?title=Packed_storage_matrix&oldid=549101766 [13 June 2019].

Appendix A

Assuming a **G** (genomic relationship) matrix available on n number of genotypes in an **M** genotype matrix, before expanding **G** for new genotypes, it needs to be corrected to **G*** (same size as **G**) for allele frequencies ($\boldsymbol{p}$) to be changed to $\boldsymbol{p}^*$ due to the new genotypes. The aim is to obtain regression coefficients α and b based on $\boldsymbol{p}$ and $\boldsymbol{p}^*$ to predict (a regression, not an exact estimation) **G*** as **G** if $\boldsymbol{p}^*$ was used instead of $\boldsymbol{p}$ (p_k = allele frequency for marker k in $\boldsymbol{p}$).

$$\mathbf{G}^* = b\mathbf{G} + \mathbf{1}\mathbf{1}'\alpha \Rightarrow \mathbf{1}\mathbf{1}'\alpha = \mathbf{G}^* - b\mathbf{G} \Rightarrow \alpha = \left(\sum_i \sum_j {G_{ij}}^* - b\sum_i \sum_j G_{ij}\right) \Big/ n^2$$

$\mathbf{G} = c\mathbf{Z}\mathbf{Z}'$, $c = 1/[2\Sigma p_k(1-p_k)]$, $\mathbf{G}^* = c^*\mathbf{Z}^*\mathbf{Z}^{*\prime}$, and $c^* = 1/[2\sum {p_k}^*(1-{p_k}^*)]$. Thus,

$$\alpha = \left[c^* \sum_i \sum_j (\mathbf{Z}^*\mathbf{Z}^{*\prime})_{ij} - bc \sum_i \sum_j (\mathbf{Z}\mathbf{Z}')_{ij}\right] \Big/ n^2$$

$\sum_i \sum_j {G_{ij}}^* = \sum_i \sum_j G_{ij}$, to obtain the unknown independent **G***, it is assumed that the slope of **G*** on **G** is due to the applied coefficients (i.e., $b = c^*/c$). Thus,

$$\alpha = c^*\left[\sum_i \sum_j (\mathbf{Z}^*\mathbf{Z}^{*\prime})_{ij} - \sum_i \sum_j (\mathbf{Z}\mathbf{Z}')_{ij}\right] \Big/ n^2$$

$$\mathbf{Z} = \mathbf{M} - 2\mathbf{P} \Rightarrow \sum_i \sum_j (\mathbf{Z}\mathbf{Z}')_{ij} = \sum_i \sum_j [(\mathbf{M}-2\mathbf{P})(\mathbf{M}-2\mathbf{P})']_{ij}$$

$$= \sum_i \sum_j (\mathbf{M}\mathbf{M}')_{ij} - 2\sum_i \sum_j (\mathbf{M}\mathbf{P}')_{ij} - 2\sum_i \sum_j (\mathbf{P}\mathbf{M}')_{ij} + 4\sum_i \sum_j (\mathbf{P}\mathbf{P}')_{ij}$$

$$= \sum_i \sum_j (\mathbf{M}\mathbf{M}')_{ij} - 4\sum_i \sum_j (\mathbf{P}\mathbf{M}')_{ij} + 4\sum_i \sum_j (\mathbf{P}\mathbf{P}')_{ij}$$

$$= \mathbf{1}_n'\mathbf{M}\mathbf{M}'\mathbf{1}_n - 4\times\mathbf{1}_n'\mathbf{P}\mathbf{M}'\mathbf{1}_n + 4\times\mathbf{1}_n'\mathbf{P}\mathbf{P}'\mathbf{1}_n$$

$$\mathbf{P} = \mathbf{1}\boldsymbol{p}' \Rightarrow \sum_i \sum_j (\mathbf{Z}\mathbf{Z}')_{ij} = \mathbf{1}_n'\mathbf{M}\mathbf{M}'\mathbf{1}_n - 4n\boldsymbol{p}'\mathbf{M}'\mathbf{1}_n + 4n^2\boldsymbol{p}'\boldsymbol{p}$$

Similarly, $\mathbf{Z}^* = \mathbf{M} - 2\mathbf{P}^*$ and $\mathbf{P}^* = \mathbf{1}\boldsymbol{p}^{*\prime}$. Therefore,

$$\sum_i \sum_j (\mathbf{Z}^*\mathbf{Z}^{*\prime})_{ij} - \sum_i \sum_j (\mathbf{Z}\mathbf{Z}')_{ij} = 4n^2[\boldsymbol{p}^{*\prime}\boldsymbol{p}^* - \boldsymbol{p}'\boldsymbol{p} - (\boldsymbol{p}^* - \boldsymbol{p})'\mathbf{M}'\mathbf{1}_n/n]$$

Downloaded From: https://bioone.org/journals/Canadian-Journal-of-Animal-Science on 28 Mar 2024
Terms of Use: https://bioone.org/terms-of-use

Pagination not final (cite DOI) / Pagination provisoire (citer le DOI)

Substituting the above line in the formula for α gives

$$\alpha = 4c^*[\boldsymbol{p}^{*\prime}\boldsymbol{p}^* - \boldsymbol{p}'\boldsymbol{p} - (\boldsymbol{p}^* - \boldsymbol{p})'\mathbf{M}'\mathbf{1}_n/n]$$

Appendix B

R function for parallel processing of matrix multiplication:

```
library("parallel")
mmultpar <- function(A, B, ncl) {
  if(ncol(A)!=nrow(B)) stop("ERROR: Dimension mis-match")
  if(ncl < 1) stop("ERROR: ncl should be a positive integer.")
  if((ncl < nrow(A)) & (ncl < ncol(B))) {
    cl = makeCluster(ncl)
    Alist = lapply(splitIndices(nrow(A), length(cl)), function(x) A[x,,drop=FALSE])
    ans = clusterApply(cl, Alist, get("%*%"), B)
    return(do.call(rbind, ans))
  } else if (nrow(A) > ncol(B)) {
    cl = makeCluster(ncol(B))
    Blist = lapply(1:ncol(B), function(x) t(B)[x,,drop=FALSE])
    ans = clusterApply(cl, Blist, get("%*%"), t(A))
    return(t(do.call(rbind, ans)))
  } else {
    cl = makeCluster(nrow(A))
    Alist = lapply(1:nrow(A), function(x) A[x,,drop=FALSE])
    ans = clusterApply(cl, Alist, get("%*%"), B)
    return(do.call(rbind, ans))
  }
}
```

Three objects are required: (*i*) the left-side matrix (`A`), (*ii*) the right-side matrix (`B`), and (*iii*) the user-defined number of cluster nodes (`ncl`).

The processes are independent, and `ncl` is not limited to the number of available nodes on the computer. If `ncl > max(nrow(A), ncol(B))`, `min(nrow(A), ncol(B))` is applied as the number of cluster nodes.

Downloaded From: https://bioone.org/journals/Canadian-Journal-of-Animal-Science on 28 Mar 2024
Terms of Use: https://bioone.org/terms-of-use