Technical note: Avoiding the direct inversion of the numerator relationship matrix for genotyped animals in single-step genomic best linear unbiased prediction solved with the preconditioned conjugate gradient

Y. Masuda, I. Misztal, Andres Legarra, S. Tsuruta, D. A. L. Lourenco, B. O. Fragomeni, I. Aguilar

Published February 2, 2017

Technical note: Avoiding the direct inversion of the numerator relationship matrix for genotyped animals in single-step genomic best linear unbiased prediction solved with the preconditioned conjugate gradient[1]

Y. Masuda,*[2] I. Misztal,* A. Legarra,† S. Tsuruta,* D. A. L. Lourenco,* B. O. Fragomeni,* and I. Aguilar‡

*Department of Animal and Dairy Science, University of Georgia, Athens 30602; †INRA, UR631 SAGA, BP 52627, 31326 Castanet-Tolosan Cedex, France; and ‡Instituto Nacional de Investigación Agropecuaria, Canelones, Uruguay 90200

ABSTRACT: This paper evaluates an efficient implementation to multiply the inverse of a numerator relationship matrix for genotyped animals ($\mathbf{A}_{22}^{-1}$) by a vector ($\mathbf{q}$). The computation is required for solving mixed model equations in single-step genomic BLUP (ssGBLUP) with the preconditioned conjugate gradient (PCG). The inverse can be decomposed into sparse matrices that are blocks of the sparse inverse of a numerator relationship matrix ($\mathbf{A}^{-1}$) including genotyped animals and their ancestors. The elements of $\mathbf{A}^{-1}$ were rapidly calculated with the Henderson's rule and stored as sparse matrices in memory. Implementation of $\mathbf{A}_{22}^{-1}\mathbf{q}$ was by a series of sparse matrix–vector multiplications. Diagonal elements of $\mathbf{A}_{22}^{-1}$, which were required as preconditioners in PCG, were approximated with a Monte Carlo method using 1,000 samples. The efficient implementation of $\mathbf{A}_{22}^{-1}\mathbf{q}$ was compared with explicit inversion of $\mathbf{A}_{22}$ with 3 data sets including about 15,000, 81,000, and 570,000 genotyped animals selected from populations with 213,000, 8.2 million, and 10.7 million pedigree animals, respectively. The explicit inversion required 1.8 GB, 49 GB, and 2,415 GB (estimated) of memory, respectively, and 42 s, 56 min, and 13.5 d (estimated), respectively, for the computations. The efficient implementation required <1 MB, 2.9 GB, and 2.3 GB of memory, respectively, and <1 sec, 3 min, and 5 min, respectively, for setting up. Only <1 sec was required for the multiplication in each PCG iteration for any data sets. When the equations in ssGBLUP are solved with the PCG algorithm, $\mathbf{A}_{22}^{-1}$ is no longer a limiting factor in the computations.

Key words: computation, genomic selection, inversion, numerator relationship matrix, preconditioned conjugate gradient, sparse matrix

 J. Anim. Sci. 2017.95:49–52
doi:10.2527/jas2016.0699

INTRODUCTION

Single-step genomic BLUP (**ssGBLUP**) is a unified approach for genomic evaluation to combine phenotypes, pedigree, and genotypes (Aguilar et al., 2010). Mixed model equations (**MME**) in ssGBLUP require the inverse of a genomic relationship matrix ($\mathbf{G}^{-1}$; VanRaden, 2008) and the inverse of a numerator relationship matrix ($\mathbf{A}_{22}^{-1}$) for genotyped animals. When the number of genotyped animals is limited, possibly less than 100,000, both inverses can be efficiently calculated (VanRaden, 2008; Aguilar et al., 2011; Fragomeni et al., 2015). When more animals are genotyped, $\mathbf{G}-1$ can be obtained efficiently using the "algorithm for proven and young" (APY; Misztal et al., 2014; Misztal, 2016), which exploits a limited rank of $\mathbf{G}$ due to a small effective population size. Faux and Gengler (2013) developed an algorithm to create $\mathbf{A}_{22}^{-1}$ directly from a pedigree. However, when the number of generations or the number of ancestors is large, the matrix becomes dense, resulting in expensive computations and more memory requirements.

Strandén and Mäntysaari (2014) showed that $\mathbf{A}_{22}^{-1}$ could be decomposed into a product of several sparse matrices. When MME are solved with the preconditioned conjugate gradient (**PCG**), the explicit $\mathbf{A}_{22}^{-1}$ is not needed because the computations require only a product of $\mathbf{A}_{22}^{-1}$ by an arbitrary vector, say $\mathbf{q}$;

[1]This research was partially funded by the United States Department of Agriculture's National Institute of Food and Agriculture (Agriculture and Food Research Initiative competitive grant 2015-67015-22936).

[2]Corresponding author: yutaka@uga.edu

Received June 5, 2016.
Accepted August 16, 2016.

that is, only $\mathbf{A}_{22}^{-1}\mathbf{q}$ is needed. With PCG, diagonals of $\mathbf{A}_{22}^{-1}$ are needed as preconditioners (Tsuruta et al., 2001). The purposes of this study were to implement efficient computations of the product $\mathbf{A}_{22}^{-1}\mathbf{q}$ and of the diagonal elements of $\mathbf{A}_{22}^{-1}$ and compare their computing costs with that of an explicit inversion.

MATERIALS AND METHODS

Indirect Multiplication of $\mathbf{A}_{22}^{-1}\mathbf{q}$

A numerator relationship matrix involving only genotyped animals and their ancestors is defined as

$$\mathbf{A}^{*} = \begin{bmatrix} \mathbf{A}_{00} & \mathbf{A}_{02} \\ \mathbf{A}_{20} & \mathbf{A}_{22} \end{bmatrix}$$

and its inverse is defined as

$$\mathbf{A}^{*-1} = \begin{bmatrix} \mathbf{A}^{00} & \mathbf{A}^{02} \\ \mathbf{A}^{20} & \mathbf{A}^{22} \end{bmatrix},$$

in which the subscript "2" represents a group of genotyped animals and "0" represents a group of their ancestors. The inverse considering inbreeding coefficients can be easily calculated using the rapid rule from a pedigree (Henderson, 1976; Quaas, 1976). The matrix $\mathbf{A}^{*-1}$ does not need either descendants of or the animals unrelated to the genotyped animals.

Using a property of the inverse (Searle, 1982, p. 260), the inverse of $\mathbf{A}_{22}$ is a function of submatrices of $\mathbf{A}^{*-1}$ (Strandén and Mäntysaari, 2014):

$$\mathbf{A}_{22}^{-1} = \mathbf{A}^{22} - \mathbf{A}^{20}\left(\mathbf{A}^{00}\right)^{-1}\mathbf{A}^{02}.$$

A product of the inverse by a vector, $\mathbf{q}$, is expressed as

$$\mathbf{A}_{22}^{-1}\mathbf{q} = \mathbf{A}^{22}\mathbf{q} - \left\{\mathbf{A}^{20}\left[\left(\mathbf{A}^{00}\right)^{-1}\left(\mathbf{A}^{02}\mathbf{q}\right)\right]\right\}. \quad [1]$$

Let 4 temporary vectors be $\mathbf{v}$, $\mathbf{w}$, $\mathbf{x}$, and $\mathbf{y}$; the above product can be computed by a sequence of matrix–vector multiplications: $\mathbf{v} = \mathbf{A}^{02}\mathbf{q}$, $\mathbf{w} = (\mathbf{A}^{00})^{-1}\mathbf{v}$, $\mathbf{x} = \mathbf{A}^{20}\mathbf{w}$, $\mathbf{y} = \mathbf{A}^{22}\mathbf{q}$, and $\mathbf{A}_{22}^{-1}\mathbf{q} = \mathbf{y} - \mathbf{x}$. The matrices $\mathbf{A}^{22}$, $\mathbf{A}^{20}$, and $\mathbf{A}^{02}$ are sparse and hence the matrix–vector multiplications can be efficiently computed. We do not need explicitly to compute $(\mathbf{A}^{00})^{-1}$ because the product $\mathbf{w} = (\mathbf{A}^{00})^{-1}\mathbf{v}$ can be computed by solving the sparse equation $\mathbf{A}^{00}\mathbf{w} = \mathbf{v}$. The Cholesky factor of $\mathbf{A}^{00}$ is typically required to solve the equation. The sparse matrices $\mathbf{A}^{22}$, $\mathbf{A}^{20}$, and $\mathbf{A}^{02}$ and the Cholesky factor of $\mathbf{A}00$ are calculated and stored in memory before the PCG iterations. The product $\mathbf{A}_{22}^{-1}\mathbf{q}$ will be calculated in each round because $\mathbf{q}$ is a vector of the current solution or search direction, and it changes every round.

Diagonals of $\mathbf{A}_{22}^{-1}$**.** The PCG algorithm requires the diagonal elements of $\mathbf{A}_{22}^{-1}$ as a part of a preconditioner. Although the explicit computation of the diagonal elements is expensive, the diagonals can be approximated by a Monte Carlo method (García-Cortés, 1994; Dong and Liu, 1994; García-Cortés and Cabrillo, 2005) as

$$\mathrm{diag}\left(\mathbf{A}_{22}^{-1}\right) \approx (1/n)\sum_{i=1}^{n}\mathbf{s}_i \odot \left(\mathbf{A}_{22}^{-1}\mathbf{s}_i\right),$$

in which n is the number of samples, $\odot$ is the direct product operator, and $\mathbf{s}_i$ is a vector of random numbers containing either +1 or −1 with equal probability and the product ($\mathbf{A}_{22}^{-1}\mathbf{s}_i$) is computed with Eq. [1]. The computation of diag($\mathbf{A}_{22}^{-1}$) has to be performed only once before the PCG iterations. Small errors in the preconditioner affect only the convergence rate in PCG, not the solutions of MME. In our tests, 1,000 samples were always sufficient to provide the same convergence rate and the solutions compared with the exact values of diag($\mathbf{A}_{22}^{-1}$).

Table 1. Description of data used in this study

Item	Data 1	Data 2	Data 3
Number of animals			
Genotyped	15,723	80,993	569,404
In selected pedigree[1]	16,694	375,946	1,436,112
In whole pedigree	213,297	8,234,208	10,710,380
Equations			
Order	219,226	8,526,614	21,985,710

[1]Genotyped animals and their ancestors.

Data

Three data sets were used for testing (Table 1). Data 1 consisted of 15,723 genotyped animals and 16,694 pedigree animals in a commercial broiler population (Lourenco et al., 2015a). Data 2 consisted of 80,993 genotyped animals and 375,946 pedigree animals in the U.S. Angus population (Lourenco et al., 2015b). Data 3 consisted of 569,404 genotyped animals and 1,436,112 pedigree animals in the U.S. Holstein population (Masuda et al., 2016).

Implementation and Computations

The inverse of $\mathbf{A}$ was calculated using the rapid method by Quaas (1976) and stored as a sparse matrix using the SPARSEM module (Misztal, 1999), as present in the blupf90 package (Misztal et al., 2016; http://nce.ads.uga.edu/wiki/BLUPmanual). The system of equations involving $\mathbf{A}^{00}$ were solved with a sparse matrix package, YAMS (Masuda et al., 2014), which used optimized dense-matrix subroutines from the Intel Math Kernel Library (Intel Corporation, Santa Clara, CA). For comparison, $\mathbf{A}_{22}$ was explicitly created and inverted as in Aguilar et al. (2011). The diagonals of

Table 2. Wall-clock time and storage memory required for the preparation of sparse components[1] of the inverse of a subset of numerator relationship matrix ($\mathbf{A}_{22}$) for genotyped animals and the results from the computations with the direct inverse of $\mathbf{A}_{22}$ for comparisons

Item	Data 1	Data 2	Data 3
Wall-clock time for preparation[2]			
Indirect approach			
Setting up $\mathbf{A}^{00}$, $\mathbf{A}^{20}$, $\mathbf{A}^{02}$, and $\mathbf{A}^{22}$	<1 s	<1 s	2 s
Factorization of $\mathbf{A}^{00}$	<1 s	167 s	125 s
Computing diagonals[3] of $\mathbf{A}_{22}^{-1}$	<1 s	148 s	134 s
Direct approach			
Setting up $\mathbf{A}_{22}$	<1 s	199 s	2.7 h[4]
Inversion of $\mathbf{A}_{22}$	42 s	56 min	13.4 d[4]
Required memory for storage			
Indirect approach	<1 MB	2.9 GB	2.3 GB
Direct approach	1.8 GB	49.0 GB	2,415 GB[4]

[1]Submatrices of the inverse of a numerator relationship matrix $\mathbf{A}^{*-1} = \begin{bmatrix} \mathbf{A}^{00} & \mathbf{A}^{02} \\ \mathbf{A}^{20} & \mathbf{A}^{22} \end{bmatrix}$, in which the subscript "2" and "0" represent a group of genotyped animals and their ancestors, respectively.

[2]Parallel processed with 8 computing cores.

[3]Using a Monte Carlo method with 1,000 samples.

[4]Extrapolated.

$\mathbf{A}_{22}^{-1}$ were calculated with the exact inverse and with the Monte Carlo method. All programs were written in Fortran 95 and compiled with the Intel Fortran Compiler 13.1 (Intel Corporation). Analyses were performed on a computer running Linux (x86_64) with Intel Xeon E5-2689 central processing unit (2.9 GHz) processor with 8 cores.

RESULTS AND DISCUSSION

Computing time and memory requirement for the explicit calculation of $\mathbf{A}_{22}$ and its inverse are shown in Table 2. Memory requirements were 1.8 GB for Data 1 and 49 GB for Data 2. Computing time was 42 s for Data 1 and 56 min for Data 2. For Data 3, the memory requirement was too much to allocate for the available computer. Extrapolating from Data 2 assuming a quadratic cost for memory and cubic costs for computing time, the computing for Data 3 would require 2,415 GB and more than 13 d to finish the inversion.

Table 2 also presents the computing time and memory requirement to prepare the sparse components of $\mathbf{A}_{22}^{-1}$. Total wall-clock time for the preparation of sparse components was short for all the data sets (less than 1 s in Data 1, 5.3 min in Data 2, and 4.4 min in Data 3). The required memory for the sparse components was much less than for the full inverse. Although Data 2 had fewer genotyped animals than Data 3, longer running time and more memory were required for the preparation. This was caused by more nonzero elements in the factor of $\mathbf{A}^{00}$, as almost all of them were newly created as "fill-in" during the factorization (Masuda et al., 2014). This illustrates the fact that $\mathbf{A}_{22}^{-1}$ can be relatively dense for the limited number of animals. Therefore, as shown in this study, the indirect computation of $\mathbf{A}_{22}^{-1}\mathbf{q}$ takes advantage of sparsity in storage and computing time compared with the direct inversion of $\mathbf{A}_{22}$. The factor of $\mathbf{A}^{00}$ as used with YAMS actually occupied more than 99% of the required memory, but this amount of memory is still negligible with current computers.

Table 3. Wall-clock time for the indirect multiplication of the inverse of a subset of numerator relationship matrix ($\mathbf{A}_{22}$) with a vector ($\mathbf{q}$) in an iteration in the preconditioned conjugate gradient (PCG) and the results from the computations with the full inverse for comparisons

Wall-clock time for $\mathbf{A}_{22}^{-1}\mathbf{q}$ per PCG iteration[1]	Data 1	Data 2	Data 3
Indirect approach	<1 s	<1 s	<1 s
Direct approach	<1 s	<1 s	59 s[2]

[1]Parallel processed with 8 computing cores.

[2]Extrapolated.

Wall-clock time for $\mathbf{A}_{22}^{-1}\mathbf{q}$ with the sparse components for one round of the PCG iteration was also negligible (<1 s) as shown in Table 3. Solving the sparse equation $\mathbf{A}^{00}\mathbf{w} = \mathbf{v}$ was inconsequential in the computation. One iteration of PCG in ssGBLUP with Data 3 took about 12 s (Masuda et al., 2016).

In ssGBLUP runs, the use of the Monte Carlo approximation of diag($\mathbf{A}_{22}^{-1}$) with 1,000 samples resulted in the same convergence rate and solutions as with the exact diagonals (results not provided). The approximation was less expensive to compute, as it required only 2.5 min for Data 2 and 2.2 min for Data 3 compared with 56 min and 13.4 d (estimated), respectively, to compute the exact diagonals. In summary, the computation of $\mathbf{A}_{22}^{-1}\mathbf{q}$ removes computing limits from ssGBLUP as these computations take a small fraction of memory and computing time even for the largest pedigrees.

LITERATURE CITED

Aguilar, I., I. Misztal, D. L. Johnson, A. Legarra, S. Tsuruta, and T. J. Lawlor. 2010. 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

Aguilar, I., I. Misztal, A. Legarra, and S. Tsuruta. 2011. Efficient computation of the genomic relationship matrix and other matrices used in single-step evaluation. J. Anim. Breed. Genet. 128:422–428. doi:10.1111/j.1439-0388.2010.00912.x

Dong, S. J., and K. F. Liu. 1994. Stochastic estimation with Z_2 noise. Phys. Lett. B 328:130–136. doi:10.1016/0370-2693(94)90440-5

Faux, P., and N. Gengler. 2013. Inversion of a part of the numerator relationship matrix using pedigree information. Genet. Sel. Evol. 45:45. doi:10.1186/1297-9686-45-45

Fragomeni, B. O., D. A. L. Lourenco, S. Tsuruta, Y. Masuda, I. Aguilar, A. Legarra, T. J. Lawlor, and I. Misztal. 2015. Use of genomic recursions in single-step genomic BLUP with a large number of genotypes. J. Dairy Sci. 98:4090–4094. doi:10.3168/jds.2014-9125

García-Cortés, L. A. 1994. Multiple trait estimation of variance components in animal models with different design matrices. In: Proc. 6th World Congr. Genet. Appl. Livest. Prod., Guelph, Canada. Vol. 18. p. 370–373.

García-Cortés, L. A., and C. Cabrillo. 2005. A Monte Carlo algorithm for efficient large matrix inversion. http://arxiv.org/abs/cs/0412107v2. (Accessed June 5, 2016.)

Henderson, C. R. 1976. A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. Biometrics 32:69–83. doi:10.2307/2529339

Lourenco, D. A. L., B. O. Fragomeni, S. Tsuruta, I. Aguilar, B. Zumbach, R. J. Hawken, A. Legarra, and I. Misztal. 2015a. Accuracy of estimated breeding values with genomic information on males, females, or both: An example on broiler chicken. Genet. Sel. Evol. 47:56. doi:10.1186/s12711-015-0137-1

Lourenco, D. A. L., S. Tsuruta, B. O. Fragomeni, Y. Masduda, I. Aguilar, A. Legarra, J. K. Bartrand, T. S. Amen, L. Wang, D. W. Moser, and I. Misztal. 2015b. Genetic evaluation using single-step genomic BLUP in American Angus. J. Anim. Sci. 93:2653–2662. doi:10.2527/jas.2014-8836

Masuda, Y., T. Baba, and M. Suzuki. 2014. Application of supernodal sparse factorization and inversion to the estimation of (co)variance components by residual maximum likelihood. J. Anim. Breed. Genet. 131:227–236. doi:10.1111/jbg.12058

Masuda, Y., I. Misztal, S. Tsuruta, A. Legarra, I. Aguilar, D. Lourenco, B. Fragomeni, and T. J. Lawlor. 2016. Implementation of genomic recursions in single-step genomic BLUP for US Holsteins with a large number of genotyped animals. J. Dairy Sci. 99:1968–1974. doi:10.3168/jds.2015-10540

Misztal, I. 1999. Complex models, larger data, simpler computing? Interbull Bull. 20:33–42.

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

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

Misztal, I., S. Tsuruta, D. A. L. Lourenco, Y. Masuda, I. Aguilar, A. Legarra, and Z. Vitezica. 2016. Manual for BLUPF90 family of programs. http://nce.ads.uga.edu/wiki/doku.php?id=documentation (Accessed 5 June 2016.)

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

Searle, S. R. 1982. Matrix algebra useful for statistics. John Wiley & Sons, Hoboken, NJ.

Strandén, I., and E. A. Mäntysaari. 2014. Comparison of some equivalent equations to solve single-step GBLUP. In: Proc. 10th World Congr. Genet. Appl. Livest. Prod., Vancouver, Canada. https://asas.org/docs/default-source/wcgalp-proceedings-oral/069_paper_9344_manuscript_568_0.pdf. (Accessed June 5, 2016.)

Tsuruta, S., I. Misztal, and I. Strandén. 2001. Use of the preconditioned conjugate gradient algorithm as a generic solver for mixed-model equations in animal breeding applications. J. Anim. Sci. 79:1166–1172. doi:10.2527/2001.7951166x

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