Using recursion to compute the inverse of the genomic relationship matrix

I. Misztal, Andres Legarra, I. Aguilar

J. Dairy Sci. 97:3943–3952
http://dx.doi.org/10.3168/jds.2013-7752

Using recursion to compute the inverse of the genomic relationship matrix

I. Misztal,*[1] A. Legarra,† and I. Aguilar‡
*Department of Animal and Dairy Science, University of Georgia, Athens 30602-2771
†INRA, UR631-SAGA, BP 52627, 31326 Castanet-Tolosan Cedex, France
‡Instituto Nacional de Investigación Agropecuaria, Las Brujas 90200, Uruguay

ABSTRACT

Computing the inverse of the genomic relationship matrix using recursion was investigated. A traditional algorithm to invert the numerator relationship matrix is based on the observation that the conditional expectation for an additive effect of 1 animal given the effects of all other animals depends on the effects of its sire and dam only, each with a coefficient of 0.5. With genomic relationships, such an expectation depends on all other genotyped animals, and the coefficients do not have any set value. For each animal, the coefficients plus the conditional variance can be called a genomic recursion. If such recursions are known, the mixed model equations can be solved without explicitly creating the inverse of the genomic relationship matrix. Several algorithms were developed to create genomic recursions. In an algorithm with sequential updates, genomic recursions are created animal by animal. That algorithm can also be used to update a known inverse of a genomic relationship matrix for additional genotypes. In an algorithm with forward updates, a newly computed recursion is immediately applied to update recursions for remaining animals. The computing costs for both algorithms depend on the sparsity pattern of the genomic recursions, but are lower or equal than for regular inversion. An algorithm for proven and young animals assumes that the genomic recursions for young animals contain coefficients only for proven animals. Such an algorithm generates exact genomic EBV in genomic BLUP and is an approximation in single-step genomic BLUP. That algorithm has a cubic cost for the number of proven animals and a linear cost for the number of young animals. The genomic recursions can provide new insight into genomic evaluation and possibly reduce costs of genetic predictions with extremely large numbers of genotypes.

Key words: genomic relationship matrix, recursion, genomic selection, single-step BLUP, preconditioned conjugate gradient (PCG) algorithm

INTRODUCTION

When only a fraction of animals are genotyped, a genomic relationship matrix **G** can be combined with a numerator relationship matrix **A** into a genomic-pedigree relationship matrix **H** (Legarra et al., 2009). Such a matrix is complicated, but has a simple inverse (Aguilar et al., 2010; Christensen and Lund, 2010). When the inverse of **H** is used with BLUP, the method is called single-step genomic BLUP (**ssGBLUP**). Advantages of ssGBLUP include simplicity of use (yet another BLUP), relatively high accuracy (Chen et al., 2011; Christensen et al., 2012; Gray et al., 2012), known and explicit control of biases because of different base populations in **A** and **G** as opposed to unknown properties of multistep methods (Tsuruta et al., 2011; Vitezica et al., 2011), and possible accounting for selection bias for genotyped animals (Patry and Ducrocq, 2011; VanRaden, 2012). Accuracy of ssGBLUP can be further improved by using a weighted **G** (Wang et al., 2012), which mimics Bayesian regressions.

The most expensive operation with ssGBLUP, as proposed by Aguilar et al. (2010) and Christensen and Lund (2010), is creating and then inverting **G**. Both operations have an approximately cubic cost with the number of genotypes. With efficient computing algorithms, both operations are feasible for up to 100,000 genotypes (Aguilar et al., 2011; Masuda and Suzuki, 2013). However, the US dairy industry has already collected over 400,000 Holstein genotypes; over 80% of genotypes are for animals without a BLUP evaluation, with a very slow increase in the number of genotypes for proven bulls (Council on Dairy Cattle Breeding, 2013).

Several approaches that do not require the inverse of **G** ($\mathbf{G}^{-1}$) have been proposed for ssGBLUP. Misztal et al. (2009) presented unsymmetric equations where only **H** was required. However, creating **H** directly is complicated. Legarra and Ducrocq (2012) presented

Received November 22, 2013.
Accepted February 10, 2014.
[1]Corresponding author: ignacy@uga.edu

different unsymmetric equations where inverses that were difficult to obtain were not required. Unsymmetric equations exhibited declining convergence with a larger number of genotypes (Aguilar et al., 2013), although they may be useful when a suitable preconditioner is found. Fernando et al. (2013) proposed a method where genotypes of nongenotyped animals were imputed and the final set of equations included SNP effects for all animals plus extra polygenic terms. However, the imputation is expensive; the volume of the imputed data are extremely large for big populations (up to dozens of millions of individuals for dairy cattle), and existing software is not applicable.

The cost of creating $\mathbf{A}$ by a tabular method (Emik and Terrill, 1949) is quadratic, and the cost to invert it directly is cubic. However, Henderson (1976) developed an algorithm based on recursion to obtain the inverse of $\mathbf{A}$ ($\mathbf{A}^{-1}$) directly at linear cost. Subsequently, $\mathbf{A}^{-1}$ can be computed for millions of animals in seconds. Faux et al. (2012) used Henderson's ideas and conditioned animals on a small number of relatives. However, the cost of their algorithm was higher than that by regular inversion. The purpose of our study was to determine whether recursion is useful in obtaining $\mathbf{G}^{-1}$ at a reasonable cost for a large number of genotypes.

MATERIALS AND METHODS

The method of Henderson (1976) to create $\mathbf{A}^{-1}$ directly depends on the recursion

$$u_{\mathrm{i}} = 0.5\left(u_{s_i} + u_{d_i}\right) + \varphi_i,$$

$$u_i = 0.5(u_{si} + u_{di}) + \varphi_i,$$

where u_i is the animal effect for animal i; s_i and d_i refer to the sire and dam of animal i, respectively; φ_i is Mendelian sampling; and founders of the pedigree are assumed to be unrelated. In matrix notation and with a genetic variance of σ_a^2 of 1 to simplify notation,

$$\mathbf{u} = \mathbf{P}\mathbf{u} + \Phi,\ \mathrm{var}(\Phi) = \mathbf{M},$$

and

$$\mathbf{A}^{-1} = (\mathbf{I} - \mathbf{P})'\mathbf{M}^{-1}(\mathbf{I} - \mathbf{P}) = \mathbf{T}'\mathbf{M}^{-1}\mathbf{T},$$

where $\mathbf{P}$ relates animals to parents; $\mathbf{T}$ is a triangular matrix if animals are ordered from oldest to youngest; and $\mathbf{M}$ is a diagonal matrix. Subsequently, $\mathbf{A}^{-1}$ can be created as a sum of outer products

$$\mathbf{A}^{-1} = \sum_i (\mathbf{t}'_{i,1:n}\,\mathbf{t}_{i,1:n}/m_i),$$

where $\mathbf{t}_{i,1:n}$ contains no more than 3 nonzero elements. Ignoring inbreeding, the value of m_i is (4 − number of known parents)/4. Henderson's rules are simple: when $\mathbf{u} \sim N(0, \mathbf{A}\sigma_a^2)$, animals are ordered from the oldest to the youngest, and all animals (including base animals) are included in the pedigree, $u_i \left| u_1, u_2, \ldots, u_{i-1}, u_{i+1}, \ldots, u_n = u_i \right| u_{s_i}, u_{d_i}$, or the conditional for an animal includes only its parents but not the rest of individuals in the pedigree. For instance, when only animals with records are included in $\mathbf{A}$ or older animals are conditioned on the younger animals, the conditional of u_i may involve more than 2 animals.

With genomic relationships, $\mathbf{u} \sim N(0, \mathbf{G}\sigma_a^2)$. The joint distribution of $u_1, \ldots, u_n$ can be written as

$$p(u_1, \ldots, u_n) = p(u_1)p(u_2|u_1)p(u_3|u_2, u_1)\ldots p(u_n|u_1, u_2, \ldots, u_{n-1}).$$

This decomposition is general and does not involve any particular ordering of individuals. Each of the conditional distributions can be written as

$$p\left(u_i|u_1, u_2, \ldots, u_{i-1}\right) \sim N \left[\mathbf{g}_{i,1:i-1}(\mathbf{G}_{1:i-1,1:i-1})^{-1}\mathbf{u}_{1:i-1}, g_{i,i} - \mathbf{g}_{i,1:i-1}(\mathbf{G}_{1:i-1,1:i-1})^{-1}\mathbf{g}'_{i,1:i-1}\right]$$

with $\mathbf{g}_{i,1:i-1}$ part of the ith row of $\mathbf{G}$, and with the following recursion equation:

$$u_i \mid u_1 \ldots u_{i-1} = \sum_{j=1}^{i-1} p_{ij}u_j + \varepsilon_i,$$

and

$$\mathbf{p}_{i,1:i-1} = \mathbf{g}_{i,1:i-1}(\mathbf{G}_{1:i-1:i-1})^{-1},\ \mathbf{M}_{i,i} = m_i = \mathrm{var}(\varepsilon_i) = \mathrm{g}_{i,i} - \mathbf{p}_{i,1:i-1}\mathbf{g}'_{i,1:i-1}. \quad [1]$$

Mimicking the developments of Henderson (1976) and Quaas (1988),

$$\mathbf{G}^{-1} = (\mathbf{I} - \mathbf{P})'\mathbf{M}^{-1}(\mathbf{I} - \mathbf{P}) = \mathbf{T}'\mathbf{M}^{-1}\mathbf{T},$$

where $\mathbf{T}$ is a triangular matrix as a result of the recursions of u_i on individuals $u_1 \ldots u_{i-1}$. Then $\mathbf{G}^{-1}$ can be created as a sum of outer products as

$$\mathbf{G}^{-1} = \sum_i (\mathbf{t}'_{i,1:n}\, \mathbf{t}_{i,1:n}/m_i).$$

The term $\{i, m_i, \mathbf{p}_{1:i-1,i}\}$ defines the genomic recursion for animal i, where m_i is the genomic Mendelian sampling and $\mathbf{p}_{1:i-1,i}$ are the genomic coefficients. Similarly to rules for $\mathbf{A}^{-1}$, $\mathbf{G}^{-1}$ can be constructed as a sum of contributions for each animal. However, $\mathbf{p}$ can have up to $n-1$ nonzero elements, and the elements of $\mathbf{p}$ generally are no longer 0 or 0.5. (See the Appendix for an example on regular and genomic recursions.)

The genomic recursions are a form of Cholesky decomposition:

$$\mathbf{G}^{-1} = \mathbf{T}'\mathbf{M}\mathbf{T} = \mathbf{U}\mathbf{D}\mathbf{U}',$$

where $\mathbf{U}$ is an upper triangular matrix and $\mathbf{D}$ is a diagonal matrix. However, the genomic recursions have a clear interpretation of what the elements of $\mathbf{U}$ and $\mathbf{D}$ are, which can be useful for insight into the genomic prediction [i.e., which animals contribute the most for a given animal's genomic EBV (**GEBV**)] and for computational refinements (progressive accumulation of elements in $\mathbf{G}^{-1}$, parallelization, and possible elimination of unimportant coefficients).

Efficient Computing Algorithms

Algorithm with Sequential Updates. The preceding algorithm is expensive, as it requires repeated inversion of sections of $\mathbf{G}$. The following algorithm creates those sections directly.

1. $GI_{n\times n} = 0; GI_{1:1,1:1} = 1/g_{1,1}$; $m_1 = g_{1,1}$.
2. $i = 2$.
3. $\mathbf{p}'_{i,1:i-1} = \mathbf{GI}_{1:i-1,1:i-1}\mathbf{g}_{1:i-1,i}$.
4. $m_i = g_{i,i} - \mathbf{p}_{i,1:i-1}\mathbf{g}_{1:i-1,i}$.
5. $$\mathbf{GI}_{1:i,1:i} = \mathbf{GI}_{1:i,1:i} + \begin{bmatrix} -\mathbf{p}'_{i,1:i-1} \\ 1 \end{bmatrix} \begin{bmatrix} -\mathbf{p}_{i,1:i-1} & 1 \end{bmatrix} / m_i.$$
6. $i = i + 1$; continue to 3 until $i > n$.

Step 3 calculates the genomic coefficients, step 4 calculates the genomic Mendelian sampling, and step 5 updates $\mathbf{G}^{-1}$ for the same animal. After the last step, $\mathbf{GI} = \mathbf{G}^{-1}$. This algorithm is equivalent to that of Sherman and Morrison (1950).

When the purpose of the algorithm is only to create genomic recursions [e.g., solving with a preconditioned conjugate gradient (**PCG**) algorithm directly], some steps can be modified:

1. $GI_{n\times n} = 0; GI_{1:1,1:1} = 1/g_{1,1}$; $m_1 = g_{1,1}$.
2. $i = 2$.
3. $$\mathbf{p}_{i,1:i-1} = \sum_{j=1}^{i-1} \left(\begin{bmatrix} -\mathbf{p}'_{j,1:j-1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \left(\begin{bmatrix} -\mathbf{p}_{j,1:j-1} & 1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{g}_{1:j,i} \\ 0 \\ \vdots \\ 0 \end{bmatrix} / m_j \right) \right).$$
4. $m_i = g_{i,i} - \mathbf{p}_{i,1:i-1}\mathbf{g}_{1:i-1,i}$.
5. $i = i + 1$; continue to 3 until $i > n$.

Zeros were added above for conformity in matrix operations. In actual computing, operations on zeros are never done.

Algorithm with Forward Updates. The modified algorithm with sequential updates can be presented in a form where genomic coefficients for animal i are immediately applied to update recursions partially for the remaining animals (step 2):

1. $\mathbf{T} = \mathbf{I}$; $m_1 = g_{1,1}$; $i = 2$.
2. $\mathbf{t}_{j,1:i-1} = \mathbf{t}_{j,1:i-1} - \mathbf{t}_{i-1,1:i-1}(\mathbf{t}'_{i-1,1:i-1}\mathbf{g}'_{1:i-1,j})/m_{i-1}$, $j = i,n$.
3. $m_i = g_{i,i} + \mathbf{t}_{i,1:i-1}\mathbf{g}_{1:i-1,i}$.
4. $i = i + 1$; continue to 2 until $i > n$.

This algorithm is more suitable for vector or parallel computations.

Costs. Without additional optimization, both the sequential and forward update algorithms have cubic cost because they implement inversion for a general symmetric matrix. Costs can be reduced if many coefficients in $\mathbf{P}$ or $\mathbf{T}$ are so small that they can be set to zero with negligible effect on $\mathbf{G}^{-1}$, although the risk of propagation of numerical errors needs to be verified for large n.

In commercial analyses, the genomic relationship for each animal needs to be computed only once. If the population is expanded to include new animals, only genomic recursions for new animals need to be added. In such a case, the algorithm with sequential updates is equivalent to formulas for updates to $\mathbf{G}^{-1}$ when additional genotypes are available (Meyer et al., 2013).

Use of G^{-1} when Solving Large Systems of Equations. Solutions to mixed model equations that include $\mathbf{G}^{-1}$ can be computed without creating $\mathbf{G}^{-1}$ explicitly. Currently, a standard method to solve large BLUP equations is PCG iteration on data (Strandén and Lidauer, 1999; Tsuruta et al., 2001). For PCG computation, the left-hand side of the system of equa-

tions is not needed explicitly; however, each round of iteration requires a product of the left-hand side times a specified vector, say $\mathbf{r}$. The product $\mathbf{G}^{-1}\mathbf{r}$ can be computed as a series of vector-by-vector multiplications involving the genomic recursions,

$$\mathbf{G}^{-1}\mathbf{r} = \sum_i \left(\mathbf{t}'_{i,1:n}\, \mathbf{t}_{i,1:n}\mathbf{r}/m_i\right),$$

at computing cost $O(ns)$, where s is average length of $\mathbf{t}$. If $\mathbf{T}$ is dense, the cost is $O(n^2)$.

Algorithm when a Population Includes Proven and Young Animals

Assume that n initial animals have been proven (i.e., have phenotypes or progeny with phenotypes) and the remaining m are young. Then

$$u_i|u_1, u_2, \ldots, u_{i-1} = \sum_{j=1}^{n} p_{ij}u_j + \sum_{j=n+1}^{n+m} p_{ij}u_j + \varepsilon_i \quad \text{for } i > n.$$

In Bayesian regressions on SNP effects (e.g., Ridge regression BLUP and BLUP_SNP) and equivalent genomic BLUP, young animals do not provide any information to GEBV of other animals. This is not true for ssGBLUP because new genomic relationships may modify old pedigree relationships; however, the effect of genotypes of young animals on GEBV of other genotyped animals is likely small, especially if n is very large. Assuming that the contributions of young animals sum to 0,

$$u_i|u_1, u_2, \ldots, u_n = u_i|\mathbf{u}_{\text{proven}} = \sum_{j\in \text{proven}} p_{ij}u_j + \varepsilon_i \text{ for } i > n.$$

If p denotes proven animals, y denotes young animals, and $\mathbf{G} = \mathbf{ZZ}'/q$ as in VanRaden (2008), then

$$\mathbf{P}_y = \mathbf{G}_{yp}\mathbf{G}_{pp}^{-1} = \mathbf{Z}_y\mathbf{Z}'_p/q\ \mathbf{G}_{pp}^{-1}$$

and

$$\text{var}(\varepsilon_i)/\sigma_u^2 \approx g_{ii} - \mathbf{G}_{ip}\mathbf{G}_{pp}^{-1}\mathbf{G}_{pi} = g_{ii} - \mathbf{z}'_i\mathbf{Z}'_p\mathbf{G}_{pp}^{-1}\mathbf{Z}_p\mathbf{z}_i/q^2 = m_i,$$

with the approximation because covariances among young animals were ignored. Finally,

$$\mathbf{G}^{-1} \approx \begin{bmatrix}\mathbf{G}_{pp}^{-1} & 0\\ 0 & 0\end{bmatrix} + \begin{bmatrix}-\mathbf{G}_{pp}^{-1}\mathbf{Z}_p\mathbf{Z}'_y/\text{q}\\ \mathbf{I}\end{bmatrix}\mathbf{M}^{-1}\begin{bmatrix}-\mathbf{Z}_y\mathbf{Z}'_p/\text{q}\,\mathbf{G}_{pp}^{-1} & \mathbf{I}\end{bmatrix}.$$

The last equation does not require the expensive operation of forming sections of $\mathbf{G}$ for young animals explicitly.

If n = number of proven animals, m = number of young animals, and ℓ = number of SNP, then

$$\text{cost}[\hat{\mathbf{u}}_y = \mathbf{Z}_y(\mathbf{Z}'_p/q)\mathbf{G}_{pp}^{-1}\hat{\mathbf{u}}_p] \sim n^3 + \ell(m+n);$$

$$\text{cost}(\mathbf{G}^{-1}\mathbf{r}) \sim n^2 + 2(\ell m + \ell n + n^2);$$

$$\text{cost}(\mathbf{G}_{pp}^{-1}) \sim n^3; \text{ and}$$

$$\text{cost}(M) \sim m\ell^2 \text{ if } \mathbf{Z}'_p\mathbf{G}_{pp}^{-1}\mathbf{Z}_p \text{ is precomputed at cost } \sim n^2\ell.$$

All costs are cubic with n, quadratic or linear with ℓ, and linear with the number of young animals. If n = 50,000, ℓ = 50,000, and PCG iteration completes in 300 rounds, then adding 450,000 young animals increases computing costs associated with $\mathbf{G}^{-1}$ by 11 times. With a regular algorithm for inversion, the costs increase 1,000 times.

Equivalent Formulas for Approximate Inverse of A_{22}

Computing ssGBLUP also requires an inverse for $\mathbf{A}_{22}$ $\left(\mathbf{A}_{22}^{-1}\right)$, which is a submatrix of $\mathbf{A}$ for genotyped animals. Using notation analogous to that for $\mathbf{G}$,

$$\mathbf{A}_{22}^{-1} \approx \begin{bmatrix}\mathbf{A}_{pp}^{-1} & 0\\ 0 & 0\end{bmatrix} + \begin{bmatrix}-\mathbf{A}_{pp}^{-1}\mathbf{A}_{py}\\ \mathbf{I}\end{bmatrix}\mathbf{M}^{-1}\begin{bmatrix}-\mathbf{A}_{yp}\mathbf{A}_{pp}^{-1} & \mathbf{I}\end{bmatrix},$$

with

$$m_i \approx a_{22,ii} - \mathbf{a}_{ip}\mathbf{A}_{pp}^{-1}\mathbf{a}_{pi}.$$

The product of a section of $\mathbf{A}$ by a vector can be computed efficiently using an algorithm by Colleau (2002). In particular, that algorithm can be used to compute a large number of $\mathbf{a}_{ip}$ in parallel (Aguilar et al., 2011).

Alternately, the formulas for $\mathbf{A}_{22}^{-1}$ may be computed using the sequential update algorithm. This requires accessing elements 1 to i of $\mathbf{A}_{22}$ at a time, which can be done either by recursion or using Colleau's algorithm. When the depth of pedigree is limited, $\mathbf{A}_{22}^{-1}$ and $\mathbf{T}$ are relatively sparse (Faux and Gengler, 2013). Subsequently, the main cost of obtaining $\mathbf{A}_{22}^{-1}$ may be computing $\mathbf{A}_{22}$.

DISCUSSION

The algorithm for genomic selection based on genomic recursions offers an extra insight compared with algorithms based on estimation of SNP effects or based on **G** obtained through usual means. Methods based on estimation of SNP effects provide insight into the genetic architecture of traits. The genomic relationships allow for comparisons and, subsequently, quality control for pedigree-based relationships. They also aid in computing theoretical accuracy by inversion of the mixed model equations. The genomic recursions allow some insight into the flow of information from older to younger animals. Such perceptions may lead to modifications of recursion coefficients for a higher realized accuracy, because theoretical accuracies using traditional **G** are inflated (Su et al., 2012). Such modifications may include accounting for decay of genomic predictions over generations (Wolc et al., 2011) or for less information from imputed genotypes (Cleveland and Hickey, 2013). A great advantage of modified genomic recursions is that they result in at least semipositive definite $\mathbf{G}^{-1}$. The application of genomic recursions depends on their properties in practice. Although they are dense for small populations (not shown), their sparsity pattern with large populations is unknown.

If animals are sequentially added to a population, both the sequential and forward update algorithms provide a possibility to create genomic recursions that are computed once for every animal, which is functionally similar to ideas of Meyer et al. (2013). The genomic recursions are calculated as animals are registered. The costs of computations once the genomic recursions are known is reasonable because those pedigrees result in quadratic (or less if recursions are sparse) costs for a PCG iteration program.

The proposed algorithms were evaluated with a simulated data set of 1,500 genotypes and ssGBLUP. In the algorithm with sequential updates, setting very small elements (−0.001 to 0.001) in **P** to zero resulted in little sparsity. Setting larger elements (−0.01 to 0.01) to zero caused large errors in $\mathbf{G}^{-1}$due to accumulation of errors. However, this algorithm worked very well for $\mathbf{A}_{22}^{-1}$. When complete **P** was computed and its small elements were zeroed (−0.01 to 0.01), the accuracy of $\mathbf{G}^{-1}$ and GEBV were almost unaffected, but the sparsity level was moderate. The sparsity level increased to >60% when **G** was blended with 20% of $\mathbf{A}_{22}$. In all computations involving the algorithm for proven and young animals, the correlations of GEBV with those using the regular algorithm were >0.99. However, this testing is not extensive and its conclusions cannot by extended to real life data sets. Future, more detailed studies will address these points.

The algorithm based on decomposition for proven and young animals can reduce costs dramatically when the number of genotypes is >50,000 and potentially help with stability during iteration. In fact, evaluation with 1 million genotypes would be possible. Although GEBV computed with **G** from the proven-young algorithm are the same as with traditional **G** in GBLUP, which is not the case for ssGBLUP. In particular, genotypes of young animals can improve accuracy of EBV for their nongenotyped relatives, although the improvement is small (Christensen et al., 2012). Therefore, ssGBLUP computed using the proven-young algorithm is an approximation. With fewer coefficients in that algorithm, the numerical properties of mixed model equations are likely better. Aguilar et al. (2013) found that the convergence rate decreases with the addition of young animals. The proven-young algorithm is of interest only when the number of animals is large, (e.g., >50,000); otherwise, conventional algorithms are cost effective (Aguilar et al., 2011).

The increase in the number of animals with highly accurate evaluations (e.g., reliability of >95%) is fairly slow for dairy cattle; perhaps 1,000 such bulls are added yearly in the United States. It is unclear how the genomic accuracy of cows is affected if they are treated as young animals in the proven-young algorithm. Normally, the addition of genotypes of cows with records increases the accuracy of genomic predictions very little. For example, the mean accuracy across breeds and traits for New Zealand dairy cattle using about 7,000 male genotypes in the training population was approximately 0.70 (Harris et al., 2013). Adding 17,000 female genotypes increased that accuracy by approximately 0.02. Accuracies for cows treated as young with the proven-young algorithm in ssGBLUP may be close to that with exact **G**, but at a much lower cost.

CONCLUSIONS

The inverse of the genomic relationship matrix, which is needed for genomic evaluation, can be computed using genomic recursions. In general, the cost of obtaining those recursions depends on their sparsity pattern and needs to be determined experimentally. In a specific case where the genotyped population included proven and young animals, the approximate cost is cubic with the number of proven animals and linear with the number of young animals.

ACKNOWLEDGMENTS

Helpful discussions with P. M. VanRaden (Animal Improvement Programs Laboratory, Agricultural Research Service, USDA, Beltsville, MD) as well as S.

Tusruta, Breno Fragomeni, Daniela Lourenco, and R. Rekaya (University of Georgia, Athens) are gratefully acknowledged. We greatly appreciate very useful corrections and suggestions by the two anonymous reviewers. This research was supported by grants from Zoetis (Kalamazoo, MI), Cobb-Vantress Inc. (Siloam Springs, AR), Smithfield Premium Genetics (Rose Hill, NC), American Angus Association (St. Joseph, MO), Holstein Association USA (Brattleboro, VT), Pig Improvement Company (Hendersonville, PIC), and Binational Agricultural Research and Development (BARD) grant IS-4394-11R.

REFERENCES

Aguilar, I., A. Legarra, S. Tsuruta, and I. Misztal. 2013. Genetic evaluation using unsymmetric single step genomic methodology with large number of genotypes. Interbull Bull. 47:222–225.

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

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.

Chen, C. Y., I. Misztal, I. Aguilar, S. Tsuruta, T. H. E. Meuwissen, S. E. Aggrey, T. Wing, and W. M. Muir. 2011. Genome-wide marker-assisted selection combining all pedigree phenotypic information with genotypic data in one step: An example using broiler chickens. J. Anim. Sci. 89:23–28.

Christensen, O. F., and M. S. Lund. 2010. Genomic prediction when some animals are not genotyped. Genet. Sel. Evol. 42:2.

Christensen, O. F., P. Madsen, B. Nielsen, T. Ostersen, and G. Su. 2012. Single-step methods for genomic evaluation in pigs. Animal 6:1565–1571.

Cleveland, M. A., and J. M. Hickey. 2013. Practical implementation of cost-effective genomic selection in commercial pig breeding using imputation. J. Anim. Sci. 91:3583–3592.

Colleau, J.-J. 2002. An indirect approach to the extensive calculation of relationship coefficients. Genet. Sel. Evol. 34:409–421.

Council on Dairy Cattle Breeding. 2013. Genotypes included in evaluations by breed, chip density, presence of phenotypes (old vs. young), and evaluation year-month (cumulative). Accessed Oct. 27, 2013. https://www.cdcb.us/Genotype/cur_density.html.

Emik, L. O., and C. E. Terrill. 1949. Systematic procedures for calculating inbreeding coefficients. J. Hered. 40:51–55.

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

Faux, P., N. Gengler, and I. Misztal. 2012. A recursive algorithm for decomposition and creation of the inverse of the genomic relationship matrix. J. Dairy Sci. 95:6093–6102.

Fernando, R. L., D. Garrick, and J. C. M. Dekkers. 2013. Bayesian regression method for genomic analyses with incomplete genotype data. Page 225 in Book of Abstracts of the 64th Annual Meeting of the European Federation of Animal Science, No. 19. Wageningen Academic Publishers, Wageningen, the Netherlands.

Gray, K. A., J. P. Cassady, Y. Huang, and C. Maltecca. 2012. Effectiveness of genomic prediction on milk flow traits in dairy cattle. Genet. Sel. Evol. 44:24.

Harris, B. L., A. M. Winkelman, and D. L. Johnson. 2013. Impact of including a large number of female genotypes on genomic selection. Interbull Bull. 47:23–27.

Henderson, C. R. 1976. A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. Biometrics 32:69–93.

Legarra, A., I. Aguilar, and I. Misztal. 2009. A relationship matrix including full pedigree and genomic information. J. Dairy Sci. 92:4656–4663.

Legarra, A., and V. Ducrocq. 2012. Computational strategies for national integration of phenotypic, genomic, and pedigree data in a single-step best linear unbiased prediction. J. Dairy Sci. 95:4629–4645.

Masuda, Y., and M. Suzuki. 2013. Efficient inversion of a large genomic relationship matrix stored on a disk using a multi-core processor and graphic processing units. J. Dairy Sci. 96(Suppl. 1):622. (Abstr.)

Meyer, K., B. Tier, and H.-U. Graser. 2013. Technical note: Updating the inverse of the genomic relationship matrix. J. Anim. Sci. 91:2583–2586.

Misztal, I., A. Legarra, and I. Aguilar. 2009. Computing procedures for genetic evaluation including phenotypic, full pedigree, and genomic information. J. Dairy Sci. 92:4648–4655.

Patry, C., and V. Ducrocq. 2011. Accounting for genomic pre-selection in national BLUP evaluations in dairy cattle. Genet. Sel. Evol. 43:30.

Quaas, R. L. 1988. Additive genetic model with groups and relationships. J. Dairy Sci. 71:1338–1345.

Sherman, J., and W. J. Morrison. 1950. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. Ann. Math. Stat. 21:124–127.

Strandén, I., and M. Lidauer. 1999. Solving large mixed linear models using preconditioned conjugate gradient iteration. J. Dairy Sci. 82:2779–2787.

Su, G., P. Madsen, U. S. Nielsen, E. A. Mäntysaari, G. P. Aamand, O. F. Christensen, and M. S. Lund. 2012. Genomic prediction for Nordic Red Cattle using one-step and selection index blending. J. Dairy Sci. 95:909–917.

Tsuruta, S., I. Misztal, I. Aguilar, and T. J. Lawlor. 2011. Multiple-trait genomic evaluation of linear type traits using genomic and phenotypic data in US Holsteins. J. Dairy Sci. 94:4198–4204.

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.

VanRaden, P. M. 2008. Efficient methods to compute genomic predictions. J. Dairy Sci. 91:4414–4423.

VanRaden, P. M. 2012. Avoiding bias from genomic pre-selection in converting daughter information across countries. Interbull Bull. 45. Accessed Mar. 21, 2014. https://journal.interbull.org/index.php/ib/article/view/1243/1241.

Vitezica, Z. G., I. Aguilar, I. Misztal, and A. Legarra. 2011. Bias in genomic predictions for populations under selection. Genet. Res. (Camb.) 93:357–366.

Wang, H., I. Misztal, I. Aguilar, A. Legarra, and W. M. Muir. 2012. Genome-wide association mapping including phenotypes from relatives without genotypes. Genet. Res. (Camb.) 94:73–83.

Wolc, A., J. Arango, P. Settar, J. E. Fulton, N. P. O'Sullivan, R. Preisinger, D. Habier, R. Fernando, D. J. Garrick, and J. C. M. Dekkers. 2011. Persistence of accuracy of genomic estimated breeding values over generations in layer chickens. Genet. Sel. Evol. 43:23.

APPENDIX

Example of Regular and Genomic Recursions

Consider 4 animals with the following pedigree:

Animal	Sire	Dam
1	0	0
2	0	0
3	0	2
4	1	2

.

The numerator relationship matrix is

$$\mathbf{A} = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.50 \\ & 1.00 & 0.50 & 0.50 \\ & & 1.00 & 0.25 \\ symm & & & 1.00 \end{bmatrix}.$$

Using equation [1] yields the following recursions:

Animal	m		P	
1	1.00			
2	1.00	0.0		
3	0.75	0.0	0.5	
4	0.50	0.5	0.5	0.0

.

Those are the same values as in the Henderson algorithm, except that the values in t are shown explicitly. They point to the following recurrence equations (setting additive variance for simplicity to 1.0):

$$\begin{aligned} u_1 &= \varepsilon_1, \mathrm{var}(\varepsilon_1) = 1.00; \\ u_2 &= \varepsilon_2, \mathrm{var}(\varepsilon_2) = 1.00; \\ u_3 &= 0.50u_2 + \varepsilon_3, \mathrm{var}(\varepsilon_3) = 0.75; \text{ and} \\ u_4 &= 0.50u_1 + 0.50u_2 + \varepsilon_4, \mathrm{var}(\varepsilon_4) = 0.50. \end{aligned}$$

If the recursion is used in the reverse order from youngest to oldest animal, the recursions are

Animal	m			P	
1	0.667	0.000	−0.333	0.000	0.667
2	0.600	0.000	0.000	0.400	0.400
3	0.937	0.000	0.000	0.000	0.250
4	1.000	0.000	0.000	0.000	0.000

,

with the new recurrence equations

$$u_1 = -0.333u_2 + 0.667u_4 + \varepsilon_1, \text{var}(\varepsilon_1) = 0.667;$$
$$u_2 = 0.4u_3 + 0.4u_4 + \varepsilon_2, \text{var}(\varepsilon_2) = 0.600;$$
$$u_3 = 0.25u_4 + \varepsilon_3, \text{var}(\varepsilon_3) = 0.937; \text{ and}$$
$$u_4 = \varepsilon_4, \text{var}(\varepsilon_4) = 1.0.$$

The coefficients with reverse ordering are no longer 0.5, although some coefficients still equal zero. The new recurrence equations have no clear meaning although they result in the same $\mathbf{A}^{-1}$.

Now assume that the genomic relationships are slightly different:

$$\mathbf{G} = \begin{bmatrix} 1.02 & -0.04 & 0.06 & 0.51 \\ & 1.05 & 0.47 & 0.54 \\ & & 0.98 & 0.31 \\ symm & & & 0.97 \end{bmatrix}.$$

Using the same formulas yields results in the following genomic recursions:

Animal	m		P	
1	1.020			
2	1.048	−0.039		.
3	0.764	0.076	0.451	
4	0.414	0.518	0.517	0.036

The new recurrence equations are

$$u_1 = \varepsilon_1, \text{var}(\varepsilon_1) = 1.02;$$
$$u_2 = -0.039u_1 + \varepsilon_2, \text{var}(\varepsilon_2) = 1.048;$$
$$u_3 = 0.076u_1 + 0.451u_2 + \varepsilon_3, \text{var}(\varepsilon_3) = 0.764; \text{ and}$$
$$u_4 = 0.414u_1 + 0.518u_2 + 0.0306u_3 + \varepsilon_4, \text{var}(\varepsilon_4) = 0.414.$$

Example for an Algorithm Based on Decomposition for Young and Proven Animals

Assume 7 animals are genotyped for 10 SNP and have the following matrix of genotypes:

$$\mathbf{Z} = \begin{bmatrix} -1 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & -1 & 0 & 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 0 & -1 & 1 & -1 & 0 & 0 & -1 \\ -1 & 0 & -1 & 1 & 1 & 0 & 1 & -1 & 1 & 1 \\ 0 & 1 & -1 & 0 & -1 & 0 & 0 & 1 & -1 & -1 \\ 1 & -1 & -1 & -1 & 0 & -1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

The genomic relationship matrix constructed according to VanRaden (2008) is

$$\mathbf{G} = \mathbf{ZZ}'/q = \begin{bmatrix} 0.795 & -0.318 & 0.000 & -0.477 & 0.636 & -0.159 & 0.318 \\ -0.318 & 0.955 & -0.159 & 0.318 & 0.000 & 0.636 & -0.477 \\ 0.000 & -0.159 & 1.114 & -0.159 & 0.159 & -0.159 & 0.000 \\ -0.477 & 0.318 & -0.159 & 0.795 & -0.477 & 0.159 & -0.318 \\ 0.636 & 0.000 & 0.159 & -0.477 & 1.273 & -0.477 & 0.159 \\ -0.159 & 0.636 & -0.159 & 0.159 & -0.477 & 0.955 & -0.159 \\ 0.318 & -0.477 & 0.000 & -0.318 & 0.159 & -0.159 & 1.114 \end{bmatrix},$$

where $q = 6.29$, which was chosen arbitrarily for a mean diagonal of 1.0. The inverse is

$$\mathbf{G}^{-1} = \begin{bmatrix} 12.229 & 14.726 & 1.704 & -2.121 & -12.225 & -12.902 & 2.114 \\ 14.726 & 23.208 & 2.269 & -4.877 & -17.428 & -19.874 & 3.996 \\ 1.704 & 2.269 & 1.191 & -0.200 & -1.817 & -1.834 & 0.426 \\ -2.121 & -4.877 & -0.200 & 3.199 & 3.930 & 4.208 & -0.530 \\ -12.225 & -17.428 & -1.817 & 3.930 & 14.774 & 15.553 & -2.742 \\ -12.902 & -19.874 & -1.834 & 4.208 & 15.553 & 18.379 & -3.225 \\ 2.114 & 3.996 & 0.426 & -0.530 & -2.742 & -3.225 & 1.786 \end{bmatrix}.$$

Assuming that observations are available for the first 5 animals,

$$\mathbf{y} = \begin{bmatrix} 31.856 & 46.657 & -6.941 & 34.636 & 51.571 & 0 & 0 \end{bmatrix},$$

the incidence matrix is

$$\mathbf{D} = \operatorname{diag}\left(\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}\right).$$

Then the GBLUP equations are

$$(\mathbf{D} + \mathbf{G}^{-1})\hat{\mathbf{u}} = \mathbf{y},$$

with solutions

$$\hat{\mathbf{u}} = \begin{bmatrix} 10.962 & 23.830 & -5.688 & 7.958 & 29.040 & 4.893 & -9.151 \end{bmatrix}'.$$

For the algorithm that decomposes the animals into proven and young,

$$\mathbf{G}^{-1} = \begin{bmatrix} 3.154 & 0.837 & 0.429 & 0.858 & -1.309 & 0 & 0 \\ 0.837 & 1.505 & 0.242 & -0.414 & -0.604 & 0 & 0 \\ 0.429 & 0.242 & 0.999 & 0.202 & -0.264 & 0 & 0 \\ 0.858 & -0.414 & 0.202 & 2.200 & 0.371 & 0 & 0 \\ -1.309 & -0.604 & -0.264 & 0.371 & 1.612 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} +$$

$$\begin{bmatrix} -0.724 & -0.123 \\ -1.008 & 0.416 \\ -0.085 & 0.085 \\ 0.259 & 0.171 \\ 0.844 & -0.010 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.080 & 0 \\ 0 & 0.820 \end{bmatrix}^{-1}$$

$$\begin{bmatrix} -0.724 & -1.008 & -0.085 & 0.259 & 0.844 & 1 & 0 \\ -0.123 & 0.416 & 0.085 & 0.171 & -0.010 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} 9.744 & 9.932 & 1.187 & -1.519 & -8.977 & -9.083 & -0.150 \\ 9.932 & 14.478 & 1.359 & -3.604 & -11.297 & -12.657 & 0.508 \\ 1.187 & 1.359 & 1.098 & -0.056 & -1.164 & -1.065 & 0.104 \\ -1.519 & -3.604 & -0.056 & 3.077 & 3.113 & 3.250 & 0.208 \\ -8.977 & -11.297 & -1.164 & 3.113 & 10.564 & 10.601 & -0.012 \\ -9.083 & -12.657 & -1.065 & 3.250 & 10.601 & 12.553 & 0 \\ -0.150 & 0.508 & 0.104 & 0.208 & -0.012 & 0 & 1.220 \end{bmatrix}.$$

Solutions from GBLUP with the new $\mathbf{G}^{-1}$ were identical.