A UNIFIED APPROACH TO UTILIZE PHENOTYPIC, FULL PEDIGREE, ANDGENOMIC INFORMATION FOR GENETIC EVALUATION

I. Misztal[1], I. Aguilar[12], A. Legarra[3], S. Tsuruta[1] D. L. Johnson[4] and T. J. Lawlor[5]

INTRODUCTION

The concept of genomic selection (Meuwissen et al., 2001) generated great excitement in the animal breeding community. With genomic information from SNP panels, one can achieve accuracy from young animals almost as high as from a progeny selection program (VanRaden et al., 2009). While costs of genotyping are still high, they are likely to drop dramatically over time. In such a case, the question of whether to use the genomic information will be replaced by how to use it efficiently.

At this time the use of genomic selection is complicated. A typical scenario would involve a multistep approach that includes 1) running a regular evaluation, 2) extracting pseudo-observations for genotyped individuals, e.g., daughter deviations or de-regressed EBV, 3) estimating SNP effects using pseudo-observations as records, and 4) possibly combining the genomic predictions with parent average (VanRaden, 2008). For smaller populations, one can run step 3 with phenotypic records; however, the information on relatives is not utilized. Step 3 usually involves estimating weights for SNP effects, mostly via BayesX procedures (Hayes et al., 2009). A procedure in which all weights are assumed equal leads to a genomic relationship matrix and is called GBLUP.

Current experiences with SNP panels of around 50-60k indicate that GBLUP is almost or as accurate as BayesX (VanRaden et al., 2009; Hayes et al., 2009). This indicates that the genomic selection works more by capturing relationships than by estimating effects of major genes. Although larger SNP panels of over 500k may capture a larger fraction of major genes, the total variance explained by those genes is likely to be small (Goldstein at al., 2009; Maher et al, 2008). Thus the primary mode of the genomic information in genetic evaluation is improved relationships among animals that also includes the information about the Mendelian sampling (Goddard, 2009).

The multi-step methodology is complicated and thus prone to errors. Also, step 3 assumes simplistic single-trait models. Misztal et al. (2009) proposed a single-step methodology

[1] Department of Animal and Dairy Science, University of Georgia, Athens, GA, United States
[2] Instituto Nacional de Investigación Agropecuaria, Las Brujas, Uruguay
[3] INRA, UR631 SAGA, BP 52627, 32326 Castanet-Tolosan, France
[4] LIC, Private Bag 3016, Hamilton 3240, New Zealand
[5] Holstein Association USA, Inc., Brattleboro, VT 05301 USA.

where step 1 is modified to use a relationships matrix that combines pedigree and genomic relationships, and steps 2-4 are eliminated. Legarra et al. (2009) and Christensen and Lund (2010) developed such a matrix, and Aguilar et al. (2010a) showed that a single step methodology is simple, fast and accurate. The purpose of this paper is to present the single step methodology with focus on efficient implementation and modification of existing software.

MATERIALS AND METHODS

Matrix H and unsymmetric equations. Assume regular mixed model equations as used in a traditional genetic evaluation, for simplicity with only a single random effect:

$$\mathbf{y}=\mathbf{Xb}+\mathbf{Zu}+\mathbf{e}$$

where **y** is a vector of records, **b** is a vector of fixed effects, and **u** is a vector of animal effects. Under a polygenic infinitesimal model of inheritance, var(**u**)=$\mathbf{A}\sigma_a^2$, where **A** is the numerator relationship matrix based on pedigree. Furthermore, var(**e**)=$\mathbf{I}\sigma_e^2$, and **X** and **Z** are appropriate incidence matrices. Misztal et al. (2009) postulated that the numerator relationship can be modified to account for genomic information:

$$\mathbf{H}= \mathbf{A}+\mathbf{A}_{\Delta}$$

where $\mathbf{A}_{\Delta}$ is a matrix that can be stored explicitly, and **H** is the new modified matrix. The regular mixed model equations (MME) are:

$$\begin{bmatrix} \boldsymbol{X'X} & \boldsymbol{X'Z} \\ \boldsymbol{Z'X} & \boldsymbol{Z'Z}+\alpha\boldsymbol{H}^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X'y} \\ \boldsymbol{Z'y} \end{bmatrix} \text{ or } \mathbf{LHS\ w} = \mathbf{RHS.}$$

Using the usual notations, **LHS**, and **RHS** are the left- and right-hand side, and **w**=$[\widehat{\boldsymbol{\beta}}' \quad \widehat{\boldsymbol{u}}']'$.

Assume that **H** is too large to be inverted. Henderson (1984) described an unsymmetric set of mixed model equations where only **H**, not necessarily of full rank, is required:

$$\begin{bmatrix} \boldsymbol{X'X} & \boldsymbol{X'Z} \\ \boldsymbol{H\ \ Z'X} & \boldsymbol{H\ \ Z'Z}+\boldsymbol{\alpha I} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X'y} \\ \boldsymbol{H\ \ Z'y} \end{bmatrix} \text{ or } \mathbf{LHS}_{\mathrm{M}}\ \mathbf{w} = \mathbf{RHS}_{\mathrm{M}}$$

For large pedigrees such a system cannot be created explicitly. Assume that the system of equations is solved using an algorithm that does not require the elements of **LHS** explicitly but only its product by a vector, say **LHS q**, as in the Preconditioned Conjugate Gradient (**PCG**) iteration on data (Tsuruta et al., 2001). With the regular equations:

$$\boldsymbol{LHS} \quad \boldsymbol{q} = \begin{bmatrix} \boldsymbol{X'Xq_1}+\boldsymbol{X'Zq_2} \\ \boldsymbol{Z'Xq_1}+\boldsymbol{Z'Zq_2}+\boldsymbol{\alpha H^{-1}q_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{c_1} \\ \boldsymbol{c_{2+}c_3}+\boldsymbol{c_4} \end{bmatrix}$$

where

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix}; \quad \boldsymbol{c}_2 = \boldsymbol{Z}'\boldsymbol{X}\boldsymbol{q}_1 + \boldsymbol{Z}'\boldsymbol{Z}\boldsymbol{q}_2; \quad \boldsymbol{c}_3 = \alpha \boldsymbol{H}^{-1}\boldsymbol{q}_2; \quad \boldsymbol{RHS} \ = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{bmatrix}.$$

With the unsymmetric equations:

$$\boldsymbol{LHS_M}\ \boldsymbol{q} = \begin{bmatrix} \boldsymbol{X'Xq_1} + \boldsymbol{X'Zq_2} \\ \boldsymbol{HZ'Xq_1} + \boldsymbol{HZ'Zq_2} + \boldsymbol{\alpha q_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{c_1} \\ \boldsymbol{H}(\boldsymbol{c_{2+}c_3}) + \boldsymbol{\alpha q_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{c_1} \\ \boldsymbol{Ac} + \boldsymbol{A_\Delta c} \quad + \boldsymbol{\alpha q_2} \end{bmatrix}$$

and

$$\boldsymbol{RHS}_M = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{Hr}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{Ar}_2 + \boldsymbol{A_\Delta r}_2 \end{bmatrix}.$$

An existing program for genetic evaluation implements ***LHS q***, and that program can be converted to use the unsymmetric equations if one can compute $\mathbf{A}_\Delta\mathbf{c}_2$ and $\mathbf{Ac}_2$ at a low cost. A product of **A** by a vector can be calculated efficiently by an algorithm given by Colleau (2002), as presented below. Additional modifications to an existing program should include a solving algorithm that works with unsymmetric matrices and a preconditioner (Misztal et al., 2009).

The Colleau Algorithm. The recurrence equation for the additive effect is:

$$\mathbf{a}=\mathbf{Pa}+\boldsymbol{\varphi}$$

where **a** is a vector of animals ordered from oldest to youngest, **φ** is a diagonal matrix of Mendelian samplings, and **P** is a matrix relating animals to their parents; this matrix has at most two elements per row, both equal to 0.5. Then:

$$\text{Var}(\mathbf{a})=\mathbf{A}=(\mathbf{I}\text{-}\mathbf{P})^{-1}\mathbf{D}(\mathbf{I}\text{-}\mathbf{P})^{-1}{}'$$

where **D**=var(**φ**). Colleau (2002) showed that the product of **A** by a vector **t** can be computed in a linear time:

$$\mathbf{v}=\mathbf{At} = (\mathbf{I}\text{-}\mathbf{P})^{-1}\mathbf{D}(\mathbf{I}\text{-}\mathbf{P})^{-1}{}'\mathbf{t} = (\mathbf{I}\text{-}\mathbf{P})^{-1}\mathbf{D}\ [(\mathbf{I}\text{-}\mathbf{P})^{-1}{}'\mathbf{t}]$$

In particular, quantities $\mathbf{r}=(\mathbf{I}\text{-}\mathbf{P})^{-1}{}'\mathbf{q}$ and $\mathbf{v}=(\mathbf{I}\text{-}\mathbf{P})^{-1}\mathbf{Dr}$ can be obtained by solving $(\mathbf{I}\text{-}\mathbf{P})'\mathbf{r}=\mathbf{q}$ and $(\mathbf{I}\text{-}\mathbf{P})\mathbf{v}=\mathbf{Dr}$, each one in a single sweep because (**I**-**P**) is triangular. The scalar formulas are:

$r_i = r_i + q_i; \quad r_{si} = r_{si}+r_i/2; \quad r_{di} = r_{di}+r_i/2; \quad i=n,..,1$
$v_i = \ d_i\, r_i + (v_{si} + v_{di})/2,\ i=1,..,n$

where s_i and d_i are positions of the sire and dam of animal i, respectively.

The Colleau (2002) algorithm can be used to compute products of sections of matrices. For instance, the products below show how to compute $\mathbf{A}_{11}\mathbf{q}$, $\mathbf{A}_{22}\mathbf{q}$, $\mathbf{A}_{21}\mathbf{q}$, or $\mathbf{A}_{22}\mathbf{q}$.

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{11}\boldsymbol{q} \\ \boldsymbol{A}_{21}\boldsymbol{q} \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{12}\boldsymbol{q} \\ \boldsymbol{A}_{22}\boldsymbol{q} \end{bmatrix}.$$

Matrix that combines pedigree and genomic relationships. Legarra et al (2009) developed matrix **H** that combines pedigree and genomic relationships. Denote:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \qquad \mathbf{A}^{-1} = \begin{bmatrix} \mathbf{A}^{11} & \mathbf{A}^{12} \\ \mathbf{A}^{21} & \mathbf{A}^{22} \end{bmatrix}$$

where subscripts 1 and 2 denoted ungenotyped and genotyped animals, respectively.
The distribution of breeding values of ungenotyped animals, conditioned on breeding values of genotyped animals, is:

$$p\left(\mathbf{u}_1 | \mathbf{u}_2\right) = N\left(\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{u}_2, \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\right).$$

Let

$$Var\left(\mathbf{u}_2\right) = \mathbf{G}$$

where G is a genomic relationship matrix as in VanRaden (2008). Then:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\left(\mathbf{G} - \mathbf{A}_{22}\right)\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{G} \\ \mathbf{G}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{G} \end{bmatrix}.$$

After rearranging,

$$\mathbf{H} = \mathbf{A} + \begin{bmatrix} \mathbf{A}_{12}\mathbf{A}_{22}^{-1} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \left[\mathbf{G} - \mathbf{A}_{22}\right] \left[\mathbf{I} \quad \mathbf{I}\right] \begin{bmatrix} \mathbf{A}_{22}^{-1}\mathbf{A}_{21} & 0 \\ 0 & \mathbf{I} \end{bmatrix}.$$

For large populations the matrix $\mathbf{A}_{\Delta}$ is large. Assume that matrices **G** and $\mathbf{A}_{22}$ (matrix of additive relationships among genotyped animals) can be stored in memory explicitly. Denote:

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{A}_{12}\mathbf{A}_{22}^{-1} & 0 \\ 0 & \mathbf{I} \end{bmatrix}, \mathbf{P}_2 = \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix}, \ \mathbf{P}_3 = \left[\mathbf{G} - \mathbf{A}_{22}\right], \ \mathbf{P}_4 = \left[\mathbf{I} \quad \mathbf{I}\right], \ \mathbf{P}_5 = \begin{bmatrix} \mathbf{A}_{22}^{-1}\mathbf{A}_{21} & 0 \\ 0 & \mathbf{I} \end{bmatrix}.$$

The product of $\mathbf{A}_{\Delta}$ by a vector **t** can be calculated efficiently without creating large matrices or multiplying them explicitly:

$$\mathbf{A}_{\Delta}\mathbf{t} = (\mathbf{P}_1(\mathbf{P}_2(\mathbf{P}_3(\mathbf{P}_4(\mathbf{P}_5\mathbf{t})))))$$

where products of $\mathbf{A}_{12}$ or $\mathbf{A}_{21}$ by a vector are obtained by the Colleau (2002) algorithm.

Inverse of H and symmetric equations. Aguilar et al. and Christensen and Lund (2010)) found that the inverse of matrix **H** as above has a simple form:

$$\mathbf{H}^{-1} = \mathbf{A}^{-1} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} - \mathbf{A}_{22}^{-1} \end{bmatrix}.$$

The new formula allows for drastically simpler computations with symmetric mixed model equations. In particular, replacing $\mathbf{A}^{-1}$ with $\mathbf{H}^{-1}$ in existing software for genetic evaluation or for estimation of variance components make those programs applicable for genomic studies. All models that are supported by a given program using $\mathbf{A}^{-1}$ are also supported by the same program when using $\mathbf{H}^{-1}$. This includes multi-trait, random regression, threshold, etc.

Efficient computation of $\mathbf{H}^{-1}$ requires efficient computation of $\mathbf{G}^{-1}$ and $\mathbf{A}_{22}^{-1}$, where the last matrix is an inverse of a pedigree-based relationship matrix for genotyped animals only.

Matrix $\mathbf{A}_{22}$ can be calculated using many methods, e.g., tabular, recursive or by the Colleau algorithm. The last option proved the fastest. Let $\mathbf{q}_j$ be a vector with 1.0 in the position corresponding to the i-th row of $\mathbf{A}_{22}$ and 0 elsewhere. Then $\mathbf{A}\ \mathbf{q}_j$ computes the i-th column of $\mathbf{A}_{22}$. For efficiency, many columns can be computed at the same time.

Matrix **G** is calculated by matrix multiplication (VanRaden et al., 2009b). Let p_j be allele frequency for genotype "2" in marker j, and let m_{ij} be genotypes for i-th animal and j-th marker such that $m_{ij} = \begin{cases} 0-2p_j & \text{- homozygous 11} \\ 1-2p_j & \text{- heterozygous 12 - 21} \\ 2-2p_j & \text{- homozygous 22} \end{cases}$

so that average $m_{i.}$ is 0. Then $\mathbf{G} = \mathbf{MM}'/k$, where the scale parameter *k* is usually computed as:

$$k = 2\sum p_j(1-p_j).$$

Gene frequencies affect the mean and scale of **G**. When equal gene frequencies are assumed, averages of the diagonal or off-diagonal elements may be much larger than in $\mathbf{A}_{22}$. Scaling **G** by regression on A as in VanRaden (2008) may result in **G** not being positive definite. With current allele frequencies, the average off-diagonal elements are close to 0. Matrix **G** can be made compatible with $\mathbf{A}_{22}$ when current allele frequencies are used and when G is scaled for an average diagonal element of 1.0. (Forni et al., WCGALP2010). Expected genetic variation in a population is proportional to trace($\mathbf{A}_{22}$) – equal to 1 if no inbreeding - or trace(**G**) (Gianola et al., 2009); scaling equalizes this variation. .

When genotyped animals include clones, **G** as constructed above is singular and cannot be easily inverted. Therefore, a common strategy is to replace **G** with $\alpha\mathbf{G} + (1\text{-}\alpha)\ \mathbf{A}_{22}$, where α is close to 1.0, e.g., 0.95. From our experience, this parameter is not critical.

Computing efficiency. Both **G** and $\mathbf{A}_{22}$ are dense matrices. Computing **G** and then inverting it may be time consuming when the number of genotyped animals and markers are high. Assuming n total animals, s genotyped animals and t SNP makers, the number of arithmetic operations is proportional to ts for $\mathbf{A}_{22}$, s^2t for **G**, and t^3 for an inverse. Operations with cubic cost become prohibitively expensive for larger matrices.

Aguilar et al (2010b) looked at computing costs using alternate codes, libraries, computers and parallel processing. For SNP data of 40k markers and 6 thousand animals, an initial code to create **G** in Fortran took from 0.5 h to 4 h, depending on the processor. Simple rearrangement of the code allowed the compiler to do an automatic optimization (including vectorization) and increased the speed 4 to 15 times, depending on the processor. Use of specialized subroutines for matrix multiplication that took into account specificity of a processor (memory speed and cache size) allowed for an improvement of about 4 times. Finally, using parallel processing in a processor with 4 cores almost quadrupled the speed. Ultimately, creation of **G**, inversion of **G**, and inversion of $\mathbf{A}_{22}$ each took less than one minute of computing. Based on simulated data, an extension of simulated data to up to 30,000 animals would increase these times to about one hour. Barring numerical problems, the maximum size of such matrices treated as dense that can be computed in a day is about 100,000.

Experiences with the single step approach

Initially, the unsymmetric equations were implemented in a modified BLUP90IOD program from the BLUPF90 family of programs (Misztal et al., 2002; Tsuruta et al., 2001). Compared to the original BLUP90IOD, one round of iteration was about four times more expensive: two times because of an unsymmetric solver and two times due to additional computations (multiplication of **H** by a vector). The convergence was as good as with the original program for smaller data sets (< 100 k animals), it deteriorated as the data size was increased to about 2 million animals, above which the iteration diverged. The convergence rate was strongly dependent on type of **G**, indicating that **G** needs to be constructed in a scale compatible with **A**.

BLUP90IOD was also modified for the symmetric equations (Aguilar, 2010a). The convergence and running time of the national evaluation for the final score in Holsteins were similar to the original program. This was regardless of type of **G** used. When the model was expanded to 5 type traits, Tsuruta et al. (2010) found the convergence reduced 2 times and the computing time per round increased about 3 times. Changes in the convergence rate with multiple-trait models and dairy data seem to be sensitive to the choice of **G** and also to modifications in formulas for **H**; replacement of the formula ($\mathbf{G}^{-1}$ - $\mathbf{A}_{22}^{-1}$) by ($\mathbf{G}^{-1}$ – 0.7 $\mathbf{A}_{22}^{-1}$)

increased the accuracy and the convergence rate while decreasing the inflation of genomic EBV.

The symmetric equations were implemented in most programs of the BLUP90 family (Misztal et al., 2002) via a custom relationship matrix. Initially, the computing was extremely slow. In these programs, sparse matrices are first stored in "hash" format, and that implementation was inefficient with large dense blocks. Refinement of a "hash function" restored performance. Subsequent analyses involved data sets with up to 300,000 animals, up to 3,500 genotypes and up to 3 traits (Chen et al, 2010). Initially, estimates of variance components were different than those without the genomic information. After scaling G for an average diagonal of 1.0 and an average off-diagonal of 0.0, these estimates were similar (Forni et al., 2010).

The programs that store the mixed model equations in memory (BLUPF90, REMLF90, AIREMLF90) become slow in multiple trait models, especially with many genotypes as dense blocks of $\mathbf{H}^{-1}$ being replicated many times. In such a case, optimized Gibbs samplers (e.g., GIBBSxF90), where multiple trait equations are assembled from single-trait equations every round, become less expensive alternatives. Most of the elements of ($\mathbf{G}^{-1}$ - $\mathbf{A}_{22}^{-1}$) are very small. It may be possible that large factions of those elements can be set to zero and the resulting matrix can be stored as a sparse matrix.

Additional issues

Large chips. When large SNP chips are available, e.g., > 500k markers, computing **G** would be more expensive, however the increase is in linear time. Studies indicate that increases of accuracies past 2-20k markers are minimal (VanRaden et al., 2009; Weigel et al., 2009). One solution with large chips is preselecting SNP markers by other methods and then either using the reduced number if this results in a better accuracy or using an evenly spaced subset otherwise. In multiple traits, the reduced subset may be different for each trait. If there is a large number of major genes detected, which may occur for specific traits, those genes can be modeled separately as fixed effects.

Many genotypes. As more animals are genotyped, the number of animals may become too large to create **H** or $\mathbf{H}^{-1}$. Two strategies may be applicable. In the first one, only genotypes relevant to the current selection would be retained. Such genotypes would be mainly those of high reliability animals or recent lower-reliability animals with some records. In the second strategy, sparse versions **H** or $\mathbf{H}^{-1}$ could be used. Particularly in $\mathbf{G}^{-1}$, most elements are very small. It may be possible to eliminate most of those elements and store $\mathbf{G}^{-1}$ as sparse. It would be desirable to develop algorithms to create sparse approximations of $\mathbf{G}^{-1}$ without creating **G** in dense form. Another alternative is using the unsymmetric equations, where the inverse of **G** is not required, and either use only selected elements of **G** or calculate a product of **G** by a vector such as t as $\mathbf{M}(\mathbf{M't})/k$ at a cost of 3nt without creating **G** explicitly.

Acknowledgements

This study was partially funded by the Holstein Association, Smithfield Premium Genetics, and AFRI grants 2009-65205-05665 and 2010-65205-20366 from the USDA NIFA Animal Genome Program.

References

Aguilar, I., Misztal I., Johnson D. L. *et al* (2010a). *J. Dairy Sci*. 93: 743-752

Aguilar, I.,Misztal, I., Legarra A. *et al*. (2010b). *J. Anim. Breed. Genet*. (submitted)

Chen, C. Y., Misztal I., Aguilar I. *et al.* (2010). In *Proc 9th WCGALP.*

Christensen, O. F., and Lund, M. S. (2010). *Genet. Sel. Evol*., 42-2.

Colleau, J. J. (2002*) Genet. Sel. Evol*. 34:409–421.

Forni, S., Aguilar, I., Misztal, I. *et al.* (2010). In *Proc 9th WCGALP.*

Gianola, D., de los Campos, G., Hill, W. G. *et al* (2009). *Genetics* 183:347-363.

Goddard, M. (2009). *Genetica* 136:245–257.

Goldstein, D.B. 2009. *New Engl. J. Med.* 360(17):1696-1698.

Hayes, B. J., P. J. Bowman, A. J. Chamberlain *et al.* (2009). *J. Dairy Sci.* 92:433–443.

Henderson, C. R. (1984) Applications of Linear Models in Animal Breeding. Univ. Guelph, 1984.

Legarra, A., Aguilar, I., and Misztal I. (2009). *J. Dairy .Sci.*, 92:4656–4663.

Maher, B. 2008. Nature 456:18-21.

Meuwissen, T. H. E., B. J. Hayes, *et al.* (2001). *Genetics* 157:1819–1829.

Misztal, I., A. Legarra, and I. Aguilar. (2009). *J. Dairy Sci.* 92: 4648-4655

Misztal, I., S. Tsuruta, T. Strabel, *et al.* (2002). In *Proc7th WCGALP.* 28–07.

Tsuruta, S., I. Misztal, and I. Stranden. 2001. *J. Anim. Sci.* 79:1166–1172.

Tsuruta, S., Aguilar, I., Misztal, I. *et al.* (2010). In *Proc 9th WCGALP.*

Weigel, K. A., de los Campos, G., Gonzalez-Recio, O. *et al.* (2009). *J. Dairy Sci.*, 92: 5248-5257.

VanRaden, P. (2008). *J. Dairy Sci.*, 91:4414-4423.

VanRaden, P. M., Van Tassell, C. P., Wiggans, G. W. *et al.* (2009). *J. Dairy Sci.* 92:16–24.

A UNIFIED APPROACH TO UTILIZE PHENOTYPIC, FULL PEDIGREE, AND GENOMIC INFORMATION FOR GENETIC EVALUATION

I. Misztal[1], I. Aguilar[12], A. Legarra[3], S. Tsuruta[1] D. L. Johnson[4] and T. J. Lawlor[5]

INTRODUCTION

The concept of genomic selection (Meuwissen et al., 2001) generated great excitement in the animal breeding community. With genomic information from SNP panels, one can achieve accuracy from young animals almost as high as from a progeny selection program (VanRaden et al., 2009). While costs of genotyping are still high, they are likely to drop dramatically over time. In such a case, the question of whether to use the genomic information will be replaced by how to use it efficiently.

At this time the use of genomic selection is complicated. A typical scenario would involve a multistep approach that includes 1) running a regular evaluation, 2) extracting pseudo-observations for genotyped individuals, e.g., daughter deviations or de-regressed EBV, 3) estimating SNP effects using pseudo-observations as records, and 4) possibly combining the genomic predictions with parent average (VanRaden, 2008). For smaller populations, one can run step 3 with phenotypic records; however, the information on relatives is not utilized. Step 3 usually involves estimating weights for SNP effects, mostly via BayesX procedures (Hayes et al., 2009). A procedure in which all weights are assumed equal leads to a genomic relationship matrix and is called GBLUP.

Current experiences with SNP panels of around 50-60k indicate that GBLUP is almost or as accurate as BayesX (VanRaden et al., 2009; Hayes et al., 2009). This indicates that the genomic selection works more by capturing relationships than by estimating effects of major genes. Although larger SNP panels of over 500k may capture a larger fraction of major genes, the total variance explained by those genes is likely to be small (Goldstein at al., 2009; Maher et al, 2008). Thus the primary mode of the genomic information in genetic evaluation is improved relationships among animals that also includes the information about the Mendelian sampling (Goddard, 2009).

The multi-step methodology is complicated and thus prone to errors. Also, step 3 assumes simplistic single-trait models. Misztal et al. (2009) proposed a single-step methodology

[1] Department of Animal and Dairy Science, University of Georgia, Athens, GA, United States
[2] Instituto Nacional de Investigación Agropecuaria, Las Brujas, Uruguay
[3] INRA, UR631 SAGA, BP 52627, 32326 Castanet-Tolosan, France
[4] LIC, Private Bag 3016, Hamilton 3240, New Zealand
[5] Holstein Association USA, Inc., Brattleboro, VT 05301 USA.

where step 1 is modified to use a relationships matrix that combines pedigree and genomic relationships, and steps 2-4 are eliminated. Legarra et al. (2009) and Christensen and Lund (2010) developed such a matrix, and Aguilar et al. (2010a) showed that a single step methodology is simple, fast and accurate. The purpose of this paper is to present the single step methodology with focus on efficient implementation and modification of existing software.

MATERIALS AND METHODS

Matrix H and unsymmetric equations. Assume regular mixed model equations as used in a traditional genetic evaluation, for simplicity with only a single random effect:

$$\mathbf{y}=\mathbf{X}\mathbf{b}+\mathbf{Z}\mathbf{u}+\mathbf{e}$$

where **y** is a vector of records, **b** is a vector of fixed effects, and **u** is a vector of animal effects. Under a polygenic infinitesimal model of inheritance, var(**u**)=$\mathbf{A}\sigma_a^2$, where **A** is the numerator relationship matrix based on pedigree. Furthermore, var(**e**)=$\mathbf{I}\sigma_e^2$, and **X** and **Z** are appropriate incidence matrices. Misztal et al. (2009) postulated that the numerator relationship can be modified to account for genomic information:

$$\mathbf{H}= \mathbf{A}+\mathbf{A}_{\Delta}$$

where $\mathbf{A}_{\Delta}$ is a matrix that can be stored explicitly, and **H** is the new modified matrix. The regular mixed model equations (MME) are:

$$\begin{bmatrix} \boldsymbol{X'X} & \boldsymbol{X'Z} \\ \boldsymbol{Z'X} & \boldsymbol{Z'Z}+\alpha\boldsymbol{H}^{-1} \end{bmatrix}\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{u}} \end{bmatrix}=\begin{bmatrix} \boldsymbol{X'y} \\ \boldsymbol{Z'y} \end{bmatrix} \text{ or } \mathbf{LHS\ w = RHS.}$$

Using the usual notations, **LHS**, and **RHS** are the left- and right-hand side, and **w**=$[\widehat{\boldsymbol{\beta}}' \quad \widehat{\boldsymbol{u}}']'$.

Assume that **H** is too large to be inverted. Henderson (1984) described an unsymmetric set of mixed model equations where only **H**, not necessarily of full rank, is required:

$$\begin{bmatrix} \boldsymbol{X'X} & \boldsymbol{X'Z} \\ \boldsymbol{H}\ \ \boldsymbol{Z'X} & \boldsymbol{H}\ \ \boldsymbol{Z'Z}+\alpha\boldsymbol{I} \end{bmatrix}\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{u}} \end{bmatrix}=\begin{bmatrix} \boldsymbol{X'y} \\ \boldsymbol{H}\ \ \boldsymbol{Z'y} \end{bmatrix} \text{ or } \mathbf{LHS}_{\mathrm{M}}\ \mathbf{w} = \mathbf{RHS}_{\mathrm{M}}$$

For large pedigrees such a system cannot be created explicitly. Assume that the system of equations is solved using an algorithm that does not require the elements of **LHS** explicitly but only its product by a vector, say **LHS q**, as in the Preconditioned Conjugate Gradient (**PCG**) iteration on data (Tsuruta et al., 2001). With the regular equations:

$$\boldsymbol{LHS}\quad \boldsymbol{q}=\begin{bmatrix} \boldsymbol{X'Xq_1+X'Zq_2} \\ \boldsymbol{Z'Xq_1+Z'Zq_2+\alpha H^{-1}q_2} \end{bmatrix}=\begin{bmatrix} \boldsymbol{c_1} \\ \boldsymbol{c_{2+}c_3+c_4} \end{bmatrix}$$

where

$$\boldsymbol{q} = \begin{bmatrix}\boldsymbol{q}_1\\ \boldsymbol{q}_2\end{bmatrix}; \quad \boldsymbol{c}_2 = \boldsymbol{Z}'\boldsymbol{X}\boldsymbol{q}_1 + \boldsymbol{Z}'\boldsymbol{Z}\boldsymbol{q}_2; \quad \boldsymbol{c}_3 = \alpha \boldsymbol{H}^{-1}\boldsymbol{q}_2; \quad \boldsymbol{RHS} \;\; = \begin{bmatrix}\boldsymbol{r}_1\\ \boldsymbol{r}_2\end{bmatrix}.$$

With the unsymmetric equations:

$$\boldsymbol{LHS}_M\, \boldsymbol{q} = \begin{bmatrix}\boldsymbol{X}'\boldsymbol{X}\boldsymbol{q}_1 + \boldsymbol{X}'\boldsymbol{Z}\boldsymbol{q}_2\\ \boldsymbol{H}\boldsymbol{Z}'\boldsymbol{X}\boldsymbol{q}_1 + \boldsymbol{H}\boldsymbol{Z}'\boldsymbol{Z}\boldsymbol{q}_2 + \boldsymbol{\alpha}\boldsymbol{q}_2\end{bmatrix} = \begin{bmatrix}\boldsymbol{c}_1\\ \boldsymbol{H}(\boldsymbol{c}_{2+}\boldsymbol{c}_3) + \boldsymbol{\alpha}\boldsymbol{q}_2\end{bmatrix} = \begin{bmatrix}\boldsymbol{c}_1\\ \boldsymbol{A}\boldsymbol{c} + \boldsymbol{A}_{\Delta}\boldsymbol{c} \quad + \boldsymbol{\alpha}\boldsymbol{q}_2\end{bmatrix}$$

and

$$\boldsymbol{RHS}_M = \begin{bmatrix}\boldsymbol{r}_1\\ \boldsymbol{H}\boldsymbol{r}_2\end{bmatrix} = \begin{bmatrix}\boldsymbol{r}_1\\ \boldsymbol{A}\boldsymbol{r}_2 + \boldsymbol{A}_{\Delta}\boldsymbol{r}_2\end{bmatrix}.$$

An existing program for genetic evaluation implements ***LHS q***, and that program can be converted to use the unsymmetric equations if one can compute $\mathbf{A}_{\Delta}\mathbf{c}_2$ and $\mathbf{A}\mathbf{c}_2$ at a low cost. A product of **A** by a vector can be calculated efficiently by an algorithm given by Colleau (2002), as presented below. Additional modifications to an existing program should include a solving algorithm that works with unsymmetric matrices and a preconditioner (Misztal et al., 2009).

The Colleau Algorithm. The recurrence equation for the additive effect is:

$$\mathbf{a}=\mathbf{P}\mathbf{a}+\boldsymbol{\varphi}$$

where **a** is a vector of animals ordered from oldest to youngest, **φ** is a diagonal matrix of Mendelian samplings, and **P** is a matrix relating animals to their parents; this matrix has at most two elements per row, both equal to 0.5. Then:

$$\text{Var}(\mathbf{a})=\mathbf{A}=(\mathbf{I}\text{-}\mathbf{P})^{-1}\mathbf{D}(\mathbf{I}\text{-}\mathbf{P})^{-1'}$$

where **D**=var(**φ**). Colleau (2002) showed that the product of **A** by a vector **t** can be computed in a linear time:

$$\mathbf{v}=\mathbf{A}\mathbf{t} = (\mathbf{I}\text{-}\mathbf{P})^{-1}\mathbf{D}(\mathbf{I}\text{-}\mathbf{P})^{-1'}\mathbf{t} = (\mathbf{I}\text{-}\mathbf{P})^{-1}\mathbf{D}\;[(\mathbf{I}\text{-}\mathbf{P})^{-1'}\mathbf{t}]$$

In particular, quantities $\mathbf{r}=(\mathbf{I}\text{-}\mathbf{P})^{-1'}\mathbf{q}$ and $\mathbf{v}=(\mathbf{I}\text{-}\mathbf{P})^{-1}\mathbf{D}\mathbf{r}$ can be obtained by solving $(\mathbf{I}\text{-}\mathbf{P})'\mathbf{r}=\mathbf{q}$ and $(\mathbf{I}\text{-}\mathbf{P})\mathbf{v}=\mathbf{D}\mathbf{r}$, each one in a single sweep because (**I**-**P**) is triangular. The scalar formulas are:

$r_i = r_i + q_i; \quad r_{si} = r_{si}+r_i/2; \quad r_{di} = r_{di}+r_i/2; \quad i=n,..,1$
$v_i = d_i\, r_i + (v_{si} + v_{di})/2, \; i=1,..,n$

where s_i and d_i are positions of the sire and dam of animal i, respectively.

The Colleau (2002) algorithm can be used to compute products of sections of matrices. For instance, the products below show how to compute $\mathbf{A}_{11}\mathbf{q}$, $\mathbf{A}_{22}\mathbf{q}$, $\mathbf{A}_{21}\mathbf{q}$, or $\mathbf{A}_{22}\mathbf{q}$.

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{11}\boldsymbol{q} \\ \boldsymbol{A}_{21}\boldsymbol{q} \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{12}\boldsymbol{q} \\ \boldsymbol{A}_{22}\boldsymbol{q} \end{bmatrix}.$$

Matrix that combines pedigree and genomic relationships. Legarra et al (2009) developed matrix **H** that combines pedigree and genomic relationships. Denote:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \qquad \mathbf{A}^{-1} = \begin{bmatrix} \mathbf{A}^{11} & \mathbf{A}^{12} \\ \mathbf{A}^{21} & \mathbf{A}^{22} \end{bmatrix}$$

where subscripts 1 and 2 denoted ungenotyped and genotyped animals, respectively.
The distribution of breeding values of ungenotyped animals, conditioned on breeding values of genotyped animals, is:

$$p\left(\mathbf{u}_1 \middle| \mathbf{u}_2\right) = N\left(\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{u}_2, \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\right).$$

Let

$$Var\left(\mathbf{u}_2\right) = \mathbf{G}$$

where G is a genomic relationship matrix as in VanRaden (2008). Then:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\left(\mathbf{G} - \mathbf{A}_{22}\right)\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{G} \\ \mathbf{G}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{G} \end{bmatrix}.$$

After rearranging,

$$\mathbf{H} = \mathbf{A} + \begin{bmatrix} \mathbf{A}_{12}\mathbf{A}_{22}^{-1} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \left[\mathbf{G} - \mathbf{A}_{22}\right] \left[\mathbf{I} \quad \mathbf{I}\right] \begin{bmatrix} \mathbf{A}_{22}^{-1}\mathbf{A}_{21} & 0 \\ 0 & \mathbf{I} \end{bmatrix}.$$

For large populations the matrix $\mathbf{A}_\Delta$ is large. Assume that matrices **G** and $\mathbf{A}_{22}$ (matrix of additive relationships among genotyped animals) can be stored in memory explicitly. Denote:

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{A}_{12}\mathbf{A}_{22}^{-1} & 0 \\ 0 & \mathbf{I} \end{bmatrix}, \mathbf{P}_2 = \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix}, \ \mathbf{P}_3 = \left[\mathbf{G} - \mathbf{A}_{22}\right], \ \mathbf{P}_4 = \left[\mathbf{I} \quad \mathbf{I}\right], \ \mathbf{P}_5 = \begin{bmatrix} \mathbf{A}_{22}^{-1}\mathbf{A}_{21} & 0 \\ 0 & \mathbf{I} \end{bmatrix}.$$

The product of $\mathbf{A}_\Delta$ by a vector **t** can be calculated efficiently without creating large matrices or multiplying them explicitly:

$$\mathbf{A}_{\Delta}\mathbf{t} = (\mathbf{P}_1(\mathbf{P}_2(\mathbf{P}_3(\mathbf{P}_4(\mathbf{P}_5\mathbf{t})))))$$

where products of $\mathbf{A}_{12}$ or $\mathbf{A}_{21}$ by a vector are obtained by the Colleau (2002) algorithm.

Inverse of H and symmetric equations. Aguilar et al. and Christensen and Lund (2010)) found that the inverse of matrix **H** as above has a simple form:

$$\mathbf{H}^{-1} = \mathbf{A}^{-1} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} - \mathbf{A}_{22}^{-1} \end{bmatrix}.$$

The new formula allows for drastically simpler computations with symmetric mixed model equations. In particular, replacing $\mathbf{A}^{-1}$ with $\mathbf{H}^{-1}$ in existing software for genetic evaluation or for estimation of variance components make those programs applicable for genomic studies. All models that are supported by a given program using $\mathbf{A}^{-1}$ are also supported by the same program when using $\mathbf{H}^{-1}$. This includes multi-trait, random regression, threshold, etc.

Efficient computation of $\mathbf{H}^{-1}$ requires efficient computation of $\mathbf{G}^{-1}$ and $\mathbf{A}_{22}^{-1}$, where the last matrix is an inverse of a pedigree-based relationship matrix for genotyped animals only.

Matrix $\mathbf{A}_{22}$ can be calculated using many methods, e.g., tabular, recursive or by the Colleau algorithm. The last option proved the fastest. Let $\mathbf{q}_j$ be a vector with 1.0 in the position corresponding to the i-th row of $\mathbf{A}_{22}$ and 0 elsewhere. Then $\mathbf{A}\ \mathbf{q}_j$ computes the i-th column of $\mathbf{A}_{22}$. For efficiency, many columns can be computed at the same time.

Matrix **G** is calculated by matrix multiplication (VanRaden et al., 2009b). Let p_j be allele frequency for genotype “2” in marker j, and let m_{ij} be genotypes for i-th animal and j-th marker such that $m_{ij} = \begin{cases} 0-2p_j & \text{- homozygous 11} \\ 1-2p_j & \text{- heterozygous 12 - 21} \\ 2-2p_j & \text{- homozygous 22} \end{cases}$

so that average $m_{i.}$ is 0. Then **G** =**MM**’/k, where the scale parameter *k* is usually computed as:

$$k = 2\sum p_j(1-p_j).$$

Gene frequencies affect the mean and scale of **G**. When equal gene frequencies are assumed, averages of the diagonal or off-diagonal elements may be much larger than in $\mathbf{A}_{22}$. Scaling **G** by regression on A as in VanRaden (2008) may result in **G** not being positive definite. With current allele frequencies, the average off-diagonal elements are close to 0. Matrix **G** can be made compatible with $\mathbf{A}_{22}$ when current allele frequencies are used and when G is scaled for an average diagonal element of 1.0. (Forni et al., WCGALP2010). Expected genetic variation in a population is proportional to trace($\mathbf{A}_{22}$) – equal to 1 if no inbreeding - or trace($\mathbf{G}$) (Gianola et al., 2009); scaling equalizes this variation. .

When genotyped animals include clones, **G** as constructed above is singular and cannot be easily inverted. Therefore, a common strategy is to replace **G** with $\alpha\mathbf{G}$ + (1-α) $\mathbf{A}_{22}$, where α is close to 1.0, e.g., 0.95. From our experience, this parameter is not critical.

Computing efficiency. Both **G** and $\mathbf{A}_{22}$ are dense matrices. Computing **G** and then inverting it may be time consuming when the number of genotyped animals and markers are high. Assuming n total animals, s genotyped animals and t SNP makers, the number of arithmetic operations is proportional to ts for $\mathbf{A}_{22}$, s^2t for **G**, and t^3 for an inverse. Operations with cubic cost become prohibitively expensive for larger matrices.

Aguilar et al (2010b) looked at computing costs using alternate codes, libraries, computers and parallel processing. For SNP data of 40k markers and 6 thousand animals, an initial code to create **G** in Fortran took from 0.5 h to 4 h, depending on the processor. Simple rearrangement of the code allowed the compiler to do an automatic optimization (including vectorization) and increased the speed 4 to 15 times, depending on the processor. Use of specialized subroutines for matrix multiplication that took into account specificity of a processor (memory speed and cache size) allowed for an improvement of about 4 times. Finally, using parallel processing in a processor with 4 cores almost quadrupled the speed. Ultimately, creation of **G**, inversion of **G**, and inversion of $\mathbf{A}_{22}$ each took less than one minute of computing. Based on simulated data, an extension of simulated data to up to 30,000 animals would increase these times to about one hour. Barring numerical problems, the maximum size of such matrices treated as dense that can be computed in a day is about 100,000.

Experiences with the single step approach

Initially, the unsymmetric equations were implemented in a modified BLUP90IOD program from the BLUPF90 family of programs (Misztal et al., 2002; Tsuruta et al., 2001). Compared to the original BLUP90IOD, one round of iteration was about four times more expensive: two times because of an unsymmetric solver and two times due to additional computations (multiplication of **H** by a vector). The convergence was as good as with the original program for smaller data sets (< 100 k animals), it deteriorated as the data size was increased to about 2 million animals, above which the iteration diverged. The convergence rate was strongly dependent on type of **G**, indicating that **G** needs to be constructed in a scale compatible with **A**.

BLUP90IOD was also modified for the symmetric equations (Aguilar, 2010a). The convergence and running time of the national evaluation for the final score in Holsteins were similar to the original program. This was regardless of type of **G** used. When the model was expanded to 5 type traits, Tsuruta et al. (2010) found the convergence reduced 2 times and the computing time per round increased about 3 times. Changes in the convergence rate with multiple-trait models and dairy data seem to be sensitive to the choice of **G** and also to modifications in formulas for **H**; replacement of the formula ($\mathbf{G}^{-1}$ - $\mathbf{A}_{22}^{-1}$) by ($\mathbf{G}^{-1}$ – 0.7 $\mathbf{A}_{22}^{-1}$)

increased the accuracy and the convergence rate while decreasing the inflation of genomic EBV.

The symmetric equations were implemented in most programs of the BLUP90 family (Misztal et al., 2002) via a custom relationship matrix. Initially, the computing was extremely slow. In these programs, sparse matrices are first stored in “hash” format, and that implementation was inefficient with large dense blocks. Refinement of a “hash function” restored performance. Subsequent analyses involved data sets with up to 300,000 animals, up to 3,500 genotypes and up to 3 traits (Chen et al, 2010). Initially, estimates of variance components were different than those without the genomic information. After scaling G for an average diagonal of 1.0 and an average off-diagonal of 0.0, these estimates were similar (Forni et al., 2010).

The programs that store the mixed model equations in memory (BLUPF90, REMLF90, AIREMLF90) become slow in multiple trait models, especially with many genotypes as dense blocks of $\mathbf{H}^{-1}$ being replicated many times. In such a case, optimized Gibbs samplers (e.g., GIBBSxF90), where multiple trait equations are assembled from single-trait equations every round, become less expensive alternatives. Most of the elements of ($\mathbf{G}^{-1}$ - $\mathbf{A}_{22}^{-1}$) are very small. It may be possible that large factions of those elements can be set to zero and the resulting matrix can be stored as a sparse matrix.

Additional issues

Large chips. When large SNP chips are available, e.g., > 500k markers, computing **G** would be more expensive, however the increase is in linear time. Studies indicate that increases of accuracies past 2-20k markers are minimal (VanRaden et al., 2009; Weigel et al., 2009). One solution with large chips is preselecting SNP markers by other methods and then either using the reduced number if this results in a better accuracy or using an evenly spaced subset otherwise. In multiple traits, the reduced subset may be different for each trait. If there is a large number of major genes detected, which may occur for specific traits, those genes can be modeled separately as fixed effects.

Many genotypes. As more animals are genotyped, the number of animals may become too large to create **H** or $\mathbf{H}^{-1}$. Two strategies may be applicable. In the first one, only genotypes relevant to the current selection would be retained. Such genotypes would be mainly those of high reliability animals or recent lower-reliability animals with some records. In the second strategy, sparse versions **H** or $\mathbf{H}^{-1}$ could be used. Particularly in $\mathbf{G}^{-1}$, most elements are very small. It may be possible to eliminate most of those elements and store $\mathbf{G}^{-1}$ as sparse. It would be desirable to develop algorithms to create sparse approximations of $\mathbf{G}^{-1}$ without creating **G** in dense form. Another alternative is using the unsymmetric equations, where the inverse of **G** is not required, and either use only selected elements of **G** or calculate a product of **G** by a vector such as t as $\mathbf{M}(\mathbf{M'}\mathbf{t})/k$ at a cost of 3nt without creating **G** explicitly.

Acknowledgements

This study was partially funded by the Holstein Association, Smithfield Premium Genetics, and AFRI grants 2009-65205-05665 and 2010-65205-20366 from the USDA NIFA Animal Genome Program.

References

Aguilar, I., Misztal I., Johnson D. L. *et al* (2010a). *J. Dairy Sci*. 93: 743-752

Aguilar, I.,Misztal, I., Legarra A. *et al*. (2010b). *J. Anim. Breed. Genet*. (submitted)

Chen, C. Y., Misztal I., Aguilar I. *et al.* (2010). In *Proc 9th WCGALP.*

Christensen, O. F., and Lund, M. S. (2010). *Genet. Sel. Evol*., 42-2.

Colleau, J. J. (2002*) Genet. Sel. Evol*. 34:409–421.

Forni, S., Aguilar, I., Misztal, I. *et al.* (2010). In *Proc 9th WCGALP.*

Gianola, D., de los Campos, G., Hill, W. G. *et al* (2009). *Genetics* 183:347-363.

Goddard, M. (2009). *Genetica* 136:245–257.

Goldstein, D.B. 2009. *New Engl. J. Med.* 360(17):1696-1698.

Hayes, B. J., P. J. Bowman, A. J. Chamberlain *et al.* (2009). *J. Dairy Sci.* 92:433–443.

Henderson, C. R. (1984) Applications of Linear Models in Animal Breeding. Univ. Guelph, 1984.

Legarra, A., Aguilar, I., and Misztal I. (2009). *J. Dairy .Sci.*, 92:4656–4663.

Maher, B. 2008. Nature 456:18-21.

Meuwissen, T. H. E., B. J. Hayes, *et al.* (2001). *Genetics* 157:1819–1829.

Misztal, I., A. Legarra, and I. Aguilar. (2009). *J. Dairy Sci.* 92: 4648-4655

Misztal, I., S. Tsuruta, T. Strabel, *et al.* (2002). In *Proc7th WCGALP.* 28–07.

Tsuruta, S., I. Misztal, and I. Stranden. 2001. *J. Anim. Sci.* 79:1166–1172.

Tsuruta, S., Aguilar, I., Misztal, I. *et al.* (2010). In *Proc 9th WCGALP.*

Weigel, K. A., de los Campos, G., Gonzalez-Recio, O. *et al.* (2009). *J. Dairy Sci.*, 92: 5248-5257.

VanRaden, P. (2008). *J. Dairy Sci.*, 91:4414-4423.

VanRaden, P. M., Van Tassell, C. P., Wiggans, G. W. *et al.* (2009). *J. Dairy Sci.* 92:16–24.