Animal (2017), **11:5**, pp 731–736
doi:10.1017/S1751731116002366

animal

Invited review: efficient computation strategies in genomic selection

I. Misztal[1†] and A. Legarra[2]

[1]*Department of Animal and Dairy Science, University of Georgia, Athens, GA 30602, USA;* [2]*UMR1388 GenePhySE, INRA, Castanet Tolosan, 31326, France*

(Received 20 April 2016; Accepted 22 September 2016; First published online 21 November 2016)

The purpose of this study is review and evaluation of computing methods used in genomic selection for animal breeding. Commonly used models include SNP BLUP with extensions (BayesA, etc), genomic BLUP (GBLUP) and single-step GBLUP (ssGBLUP). These models are applied for genomewide association studies (GWAS), genomic prediction and parameter estimation. Solving methods include finite Cholesky decomposition possibly with a sparse implementation, and iterative Gauss–Seidel (GS) or preconditioned conjugate gradient (PCG), the last two methods possibly with iteration on data. Details are provided that can drastically decrease some computations. For SNP BLUP especially with sampling and large number of SNP, the only choice is GS with iteration on data and adjustment of residuals. If only solutions are required, PCG by iteration on data is a clear choice. A genomic relationship matrix (GRM) has limited dimensionality due to small effective population size, resulting in infinite number of generalized inverses of GRM for large genotyped populations. A specific inverse called APY requires only a small fraction of GRM, is sparse and can be computed and stored at a low cost for millions of animals. With APY inverse and PCG iteration, GBLUP and ssGBLUP can be applied to any population. Both tools can be applied to GWAS. When the system of equations is sparse but contains dense blocks, a recently developed package for sparse Cholesky decomposition and sparse inversion called YAMS has greatly improved performance over packages where such blocks were treated as sparse. With YAMS, GREML and possibly single-step GREML can be applied to populations with >50 000 genotyped animals. From a computational perspective, genomic selection is becoming a mature methodology.

Keywords: genomic selection, single-step, genomic relationship matrix, inverse, REML

Implications

Tools used in genomic selection range from those based on estimation of SNP effects to animal models using a genomic relationship matrix (GRM). When many animals are ungenotyped, modeling options include multi-step or single-step methods. Computations involving mixed models with SNP effects or the inverse of GRM have good computing properties, whereas those including their combination or plain GRM do not. Recent discovery of efficient inverse of GRM removed computing limits from genomic analyses involving GRM. Same discovery enables GREML for very large data sets.

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 test (Schaeffer, 2006; VanRaden *et al.*, 2009) at a greatly reduced cost. Initially the genomic computations used versions of BLUP with SNP fitted as random effects, such as SNP BLUP, BayesB, etc., (see Gianola *et al.*, 2009 for a review). An alternative form of SNP BLUP is genomic BLUP (GBLUP), where the animal effect is fit with a GRM (VanRaden, 2008). For prediction, SNP BLUP and GBLUP are equivalent models (VanRaden, 2008; Strandén and Christensen, 2011).

When only a small fraction of the population is genotyped, the information from ungenotyped animals can be summarized in pseudo-observations for genotyped animals. Alternatively, GBLUP can be extended to single-step GBLUP (ssGBLUP) (Aguilar *et al.*, 2010; Christensen and Lund, 2010), where a numerator relationship matrix (NRM) for all individuals and GRM are combined and then applied to BLUP. Benefits of ssGBLUP include simplicity of application (another BLUP), avoidance of double counting, and accounting for pre-selection on Mendelian sampling (Legarra *et al.*, 2014). ssGBLUP was

[†] E-mail: Ignacy@uga.edu

extended to be based partially or completely on SNP effects (Fernando *et al*., 2014; Liu *et al*., 2014).

Large interest in the genomic community exists in identifying an optimal set of SNP and their variances for increased accuracy of evaluation. Such an identification is straightforward with Bayesian SNP models (e.g. Gianola *et al*., 2009). As SNP BLUP and GBLUP are equivalent, SNP and their variances selected in Bayesian SNP models can be transformed to weighted GRM and subsequently BLUP. Methods exist to generate SNP weights directly in the GBLUP model (VanRaden, 2008; Zhang *et al*., 2010; Sun *et al*., 2012) or ssGBLUP (Wang *et al*., 2012).

Initially the extent of genomic selection was limited by costs of genotyping. With these costs dropping dramatically, the number of genotyped animals has increased dramatically, with very high-density SNP chips available. In dairy cattle, over 1 million Holsteins had been genotyped in the United States as of March 2016 (Council on Dairy Cattle Breeding; https://www.cdcb.us/Genotype/cur_density.html). Subsequently, computing costs are becoming more of an issue. The purpose of this paper is to present and discuss computational methods in genomic selection.

Definition of terms and basic models

Let **A** denote a NRM based on pedigrees. Let **Z** be a matrix of gene content with z_{ij} being the number of occurrences of the minor allele in SNP *j* of animal *i*. The values of 'raw' z_{ij} are 0, 1 and 2, whereas that of centered z_{ij} is $-2p_j$, $1-p_j$ and $2-p_j$, where p_j is gene frequency of SNP *j*. The simplest SNP BLUP model is

$$y_i = \mu + \sum_j z_{ij} a_j + e_i$$

where y_i is a (pseudo)phenotype of animal *i*, $\mathbf{a} \sim N(0, \mathbf{I}\sigma_a^2)$ the vector of SNP effects, $\mathbf{e} \sim N(0, \mathbf{I}\sigma_e^2)$ the vector of residuals, and σ_a^2 and σ_e^2 are SNP and residual variances, respectively. GBLUP can be defined as

$$y_i = \mu + u_i + e_i$$

where $\mathbf{u} \sim N(0, \mathbf{G}\sigma_u^2)$ is a vector of animal effects, **G** the GRM and σ_u^2 the additive variance. **G** can be derived by a transformation from SNP BLUP:

$$\mathbf{G} = \mathbf{ZZ}' \frac{\sigma_a^2}{\sigma_u^2}$$

possibly with an alternative scaling factor (VanRaden, 2008):

$$\mathbf{G} = \frac{\mathbf{ZZ}'}{\sum_j [p_j(1-p_j)]}$$

If SNP BLUP involves heterogeneous variance for SNP: $\mathbf{a} \sim N(0, \mathbf{D}\sigma_a^2)$, where **D** is a diagonal matrix of weights, the equivalent 'weighted' **G** is

$$\mathbf{G} = \frac{\mathbf{ZDZ}'}{\sum_j [p_j(1-p_j)]}$$

SNP BLUP and GBLUP are equivalent models (Strandén and Christensen, 2011).

When only a fraction of animals is genotyped, the numerator and genomic relationships can be combined (Legarra *et al*., 2009; Christensen and Lund, 2010):

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

where **H** is a combined matrix and indices in **A** refer to ungenotyped (1) and genotyped (2) animals. The inverse of matrix **H** is (Aguilar *et al*., 2010; Christensen and Lund, 2010)

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

BLUP with matrix **H** is called ssGBLUP.

Computing operations in animals breeding

Common operations in animal breeding include solving of mixed model equations and variance component estimation either via REML or Bayesian methods with Gibbs sampling. Such operations in the animal breeding context are described in Mrode (2014) and Misztal (2014), and are typically based on Henderson's mixed model equations (MME). Let this linear system of equations be

$$\mathbf{B\,x} = \mathbf{y}$$

where **B** is left-hand-side (LHS), **y** the right-hand side, and **x** the vector of solutions. Solving mixed model equations can be accomplished by finite and iterative methods. In finite methods, exact solutions (except for numerical errors) are obtained in a finite number of steps. Popular methods for general matrices are based on the LU decomposition (**B** = **LU**, where **L** and **U** are lower and upper triangular, respectively) and for symmetric matrices the Cholesky decomposition (**B** = **LL'**, where **L** is lower triangular). As often mixed model equations are not full rank, finite methods used for mixed models need to compute generalized solutions rather than cause numerical exceptions. For example, the Cholesky decomposition can be modified to

$$\mathbf{B} = \mathbf{LDL}'$$

where **D** is a diagonal matrix with zero elements for redundant equations and non-zeros otherwise. Although finite methods usually compute solutions with high precision, their cost is high, usually close to cubic. Finite methods can be used to calculate inverses, which are used in REML or for prediction error variance (PEV) calculations. In general, the cost of the inverse is a few times that of a solution by decomposition.

https://doi.org/10.1017/S1751731116002366 Published online by Cambridge University Press

When matrices are sparse, with nearly all elements of LHS equal to 0, a 'sparse' Cholesky decomposition exists that avoids operations with zeros, leading to about quadratic rather than cubic costs (George and Liu, 1981). Also, an algorithm exists for computing a 'sparse' inverse, where elements computed are those corresponding to non-zeroes in the original matrix (Takahashi *et al.*, 1973). Such an inverse is sufficient to calculate traces in REML (Misztal and Perez-Enciso, 1993) or PEV. However, equations involving SNP effects are usually dense. When the system of equations is sparse but contains dense blocks, a sparse matrix package can be modified for greatly improved performance in REML and other applications (Masuda *et al.*, 2015). This is quite new as most matrix packages are efficient for either dense or sparse matrices, but not for mixtures of both.

Iterative methods are such that progressively more accurate solutions are obtained every next round. Two such methods are popular in animal breeding. The first one is Gauss–Seidel iteration (GS), which is also a backbone of Gibbs sampling. GS is very simple, but requires access to equations by rows. The second one is preconditioned conjugate gradient (PCG), a method difficult to understand, but easy to implement and usually with superior convergence rate to GS (Tsuruta *et al.*, 2001).

In animal breeding application, the size of LHS can be much larger than the size of data to generate LHS. For example, one line of data in a 10 trait model with 10 effects per trait includes 20 numbers (one for each trait and effect) but generates 10 000 contributions to LHS. A special implementation of iteration algorithms is matrix-free or by iteration on data (Schaeffer and Kennedy, 1986; Misztal and Gianola, 1987), where coefficients of LHS are regenerated from the data every round of iteration. Although the implementation of iteration on data by GS can be complicated as the coefficients need to be recreated row by row, such an implementation in PCG is trivial as PCG only requires a product of **B** by a vector (**Bq**, with **q** a given vector), with no individual elements necessary (except for diagonals).

Tricks and rules in mixed model and genomic computations

Product of numerator relationship matrix

Computing the NRM **A** by a tabular method has a quadratic cost. Also, computing relationships for specific animals requires creating a matrix for all ancestors. For large populations, **A** would not fit into memory. When only a product **A** by a vector, say **q**, is needed, **Aq** can be computed inexpensively by two scans of the pedigree (Colleau, 2002). By selecting zeros in **q**, one can also compute a product of a section of **A** by a vector (e.g. $\mathbf{A}_{22}\,\mathbf{q}$) (Misztal *et al.*, 2009; Aguilar *et al.*, 2011).

Sequential multiplication and preconditioned conjugate gradient

The LHS of the mixed model equations contains expressions like **Z'Z**. The PCG algorithm by iteration on data requires computing **Z'Zq**, where **q** is a given vector. Strandén and Lidauer (1999) found that while the cost of (**Z'Z**)**q** is very high, the cost of **Z'**(**Zq**) is much lower.

Convergence properties with various systems of equations

When mixed models are solved by iteration (GS or PCG), the convergence rate is better with fewer equations per phenotype. Experiences indicate that the convergence rate is very good with sire models, much slower with animal models, and could be especially slow with young animals not tied to phenotypes or progeny.

The regular MME equations involve an inverse of **A** or **G**, e.g.

$$(\mathbf{I}+\mathbf{A}^{-1}\alpha)\hat{\mathbf{a}}=\mathbf{y}$$

An equivalent system of equation avoids creating an inverse (Strandén and Garrick, 2009):

$$(\mathbf{A}+\mathbf{I}\alpha)\hat{\mathbf{a}}=\mathbf{Ay}$$

The convergence rate in such models is poor and may not occur for larger systems of equations. This is because $\mathbf{A}^{-1}$ 'connects' only parents and progeny resulting in 'sparse' equations, whereas all animals in **A** are connected, resulting in dense equations. As **G** differs little from **A** (VanRaden, 2008; Wang *et al.*, 2014), the same thinking applies to models with **G** and $\mathbf{G}^{-1}$.

Computations in SNP BLUP (and BayesX)

SNP BLUP and its various forms (including BayesA, BayesB, etc.) are very popular models in genomic computations. When the number of SNP is small, the MME for SNP BLUP can be created explicitly. As the memory and compute requirements are quadratic, explicit storage of SNP BLUP equations is unfeasible with large number of SNP. For example, memory requirements for MME with 1 million SNP would be 10^{12} elements also with 10^{12} elements of MME created from each genotyped animal. Various options in such a case have been explored by Legarra and Misztal (2008) with a data set on about 2 k individuals and 20 k SNP markers. Solution by Cholesky decomposition took 2 h, and computations would increase cubically with the number of SNP, whereas memory would increase quadratically. With solution by GS and straightforward iteration on data, computations took about 4 days, with linear memory and quadratic memory costs. With a special GS implementation called GSRU, where currently computed solutions are used to adjust residuals for all individuals, computing took about 1 min and both computing and memory costs were linear. Finally, with PCG iteration on data, computing took only 20 s, with approximately linear costs for memory and computations. Although GSRU is the only realistic algorithm with millions of SNP for Bayesian models, the PCG approach can be useful for models where SNP variances are derived from SNP solutions (Zhang *et al.*, 2010; Sun *et al.*, 2012; Wang *et al.*, 2012). A hybrid method would use PCG iteration

https://doi.org/10.1017/S1751731116002366 Published online by Cambridge University Press

with inner (non-Monte Carlo-based) updates of weights (VanRaden, 2008).

Variations of single-step equations

Due to the prohibitive cubic cost of inversion of **G** with many genotyped animals, alternative equations were proposed for ssGBLUP. Equations in Misztal *et al.* (2009) used **H** rather than $\mathbf{H}^{-1}$. Convergence was achieved only with a small number of animals. Equations by Legarra and Ducrocq (2012) used **G** and $\mathbf{A}_{22}$, which are easy to use, instead of $\mathbf{G}^{-1}$ and $\mathbf{A}_{22}^{-1}$. The convergence rate was slow with medium models and not achieved with large models (Aguilar *et al.*, 2013). Liu *et al.* (2014) proposed equations with SNP effects for genotyped animals. They reported lack of convergence with real data. All these models illustrate computing advantages with inverses of dispersion matrices ($\mathbf{A}^{-1}$ or $\mathbf{G}^{-1}$). As SNP BLUP models have good computing properties, Fernando *et al.* (2014) looked into ssGBLUP where ungenotyped animals were imputed and only SNP effects (plus imputation errors for ungenotyped animals) were estimated. Although convergence rates were good, imputations required a supercomputer and imputed genotypes required large memory for big populations. Better computing times were reported using a computer with fast graphic cards (Golden *et al.*, 2016). Meuwissen *et al.* (2015) presented a new type of **H** matrix based on (possibly fractional) imputation of ancestors by segregation analyses, followed by a sophisticated tailoring of **G** to **A** to compensate for errors in imputation. Although the cost of segregation analysis is high, it is not yet clear whether such an approach offers accuracy benefits when most ancestors are genotyped, and the process of tailoring is rather complex. All non-traditional ssGBLUP would require new research and investment in code for implementation of more complex models, such as multiple-trait, random regression or maternal models.

Sparse genomic relationship matrix

Computations with ssGBLUP would be easy for large number of animals if $\mathbf{G}^{-1}$ was sparse, like $\mathbf{A}^{-1}$. Past studies suggested that **G** had a limited rank, as inversion is unstable with >5 to 10 k animals, and **G**s as used in genetic evaluations are blended with $\mathbf{A}_{22}$ (VanRaden, 2008; Aguilar *et al.* 2010) to achieve full rank. The eigenvalue decomposition of **G** is:

$$\mathbf{G} = \mathbf{UDU}'$$

where **U** is a matrix of eigenvectors and **D** the matrix of eigenvalues. If all eigenvalues are positive, the inverse of **G** is

$$\mathbf{G}^{-1} = \mathbf{U}'\mathbf{D}^{-1}\mathbf{U}$$

If some eigenvalues are 0, the inverse does not exist. If, due to noise, many eigenvalues are close to 0 and carry no information, they can cause instability of the inverse through large elements of $\mathbf{D}^{-1}$.

Let $\mathbf{D}_t$ indicate a fraction of **D** with non-negligible eigenvalues, and let $\mathbf{U}_t$ be corresponding eigenvalues. Then

$$\mathbf{G}^{-} = \mathbf{U}_t'\mathbf{D}_t^{-1}\mathbf{U}_t$$

If $\mathbf{G}^{-1}$ is to be created explicitly, the computing cost is cubic with quadratic storage. If $\mathbf{G}^{-}$ is used by a PCG algorithm as

$$\mathbf{G}^{-}\mathbf{q} = \left(\mathbf{U}_t'\left(\mathbf{D}_t^{-1}\left(\mathbf{U}_t\mathbf{q}\right)\right)\right)$$

and the number of eigenvalues in $\mathbf{D}_t$ is small, $\mathbf{G}^{-}\mathbf{q}$ can be calculated and stored at linear cost with respect to the number of genotyped animals. However, computing **U** and **D** is expensive and requires storage of **G** in full.

When **G** has a limited dimensionality, Misztal *et al.* (2014) proposed to calculate $\mathbf{G}^{-1}$ using only a fraction of **G** and without eigenvalue decomposition. Let

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{cc} & \mathbf{G}_{cn} \\ \mathbf{G}_{cn} & \mathbf{G}_{nn} \end{bmatrix}$$

where index *c* denotes core animals and index *n* non-core animals. In the algorithm for proven and young animals (APY)

$$\mathbf{G}^{-1} \approx \begin{bmatrix} \mathbf{G}_{cc}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} -\mathbf{G}_{cc}^{-1}\mathbf{G}_{cn} \\ \mathbf{I} \end{bmatrix} \mathbf{M}^{-1} \begin{bmatrix} -\mathbf{G}_{nc}\mathbf{G}_{cc}^{-1} & \mathbf{I} \end{bmatrix},$$

where **M** is a diagonal matrix with elements $m_i = g_{ii} - \mathbf{G}_{ic}\mathbf{G}_{cc}^{-1}\mathbf{G}_{ci}$ for individual *i* in the core group. Inversion of **G** by APY has a cubic cost (and quadratic memory) for core individuals and a linear cost (and linear memory) for non-core individuals. Figure 1 shows an example non-zero structure of a regular inverse (after blending to make it positive-definite) and different APY inverses. The inverses are diagonal for non-core animals. As the regular **G** is singular for a large population, an infinite number of generalized inverses exist, including those by the APY algorithm with different sets of core animals.

Core animals can be randomly selected (Fragomeni *et al.*, 2015), with an optimal number of about 12 000 for Holsteins

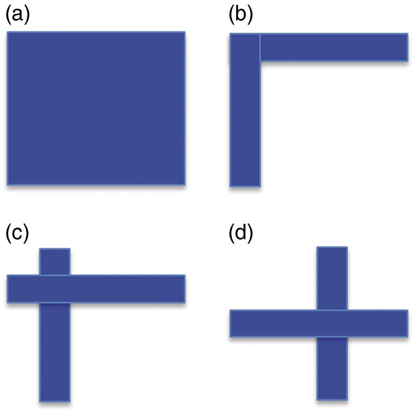

Figure 1 Non-zero pattern of a regular (a) and APY (b to d) inverses of the genomic relationship matrix with different choices of core animals.

https://doi.org/10.1017/S1751731116002366 Published online by Cambridge University Press

(Masuda *et al*., 2016) and 10 000 for Angus (Lourenco *et al*., 2015). The number of core animals is linked to effective population size (Misztal, 2016). Optimal number of core animals is equal to the number of the largest eigenvalues in **D** explaining 98% of the variance in **G** (Pocrnic *et al*., 2016a). With such a number, prediction of breeding values with APY was more accurate than with a regular inverse. Computation of $\mathbf{G}^{-1}$ for 570 k genotyped Holsteins by APY took only about 2 h, whereas a month computing with a large memory would be required for a regular inverse (Masuda *et al*., 2016). The APY algorithm removes computing limits from inversion of **G** and subsequently ssGBLUP.

Computing of $\mathbf{A}_{22}$

ssGBLUP as in Aguilar *et al.* (2010) requires $\mathbf{A}_{22}^{-1}$. Although $\mathbf{A}^{-1}$ is sparse, $\mathbf{A}_{22}^{-1}$ may be relatively dense (Faux and Gengler, 2013), and its storage for potentially millions of genotyped animals impossible. This matrix can be calculated indirectly (Strandén and Mäntysaari, 2014).

$$\mathbf{A}_{22}^{-1} = \mathbf{A}^{22} - \left(\mathbf{A}^{12}\right)'\left(\mathbf{A}^{11}\right)^{-1}\mathbf{A}^{12}$$

where all submatrices are sparse and can be stored explicitly. With PCG iteration, explicit A_{22}^{-1} is not required, and its product with a vector **q** can be calculated every round as follows (Masuda *et al*., 2016):

$$\mathbf{A}_{22}^{-1}\mathbf{q} = \left[\mathbf{A}^{22} - \left(\mathbf{A}^{12}\right)'\left(\mathbf{A}^{11}\right)^{-1}\mathbf{A}^{12}\right]\mathbf{q}$$

In particular

$$\boldsymbol{s} = \left(\mathbf{A}^{11}\right)^{-1}\mathbf{A}^{12}\mathbf{q}$$

is computed as a solution to:

$$\mathbf{A}^{11}\boldsymbol{s} = \mathbf{A}^{12}\mathbf{q}$$

As all matrices are sparse, the cost of above calculations is small (Masuda *et al*., 2016).

GREML and APY

GREML is REML using a GRM. Similarly, REML using matrix **H** can be called single-step GREML (ssGREML). In the past, the success with REML in animal populations was due to sparse inverse of $\mathbf{A}^{-1}$ and sparse matrix factorization/inversion (Misztal and Perez-Enciso, 1993; Pérez-Enciso *et al*., 1994). If **G** is dense and large, Masuda *et al*. (2015) showed that GREML with the old sparse matrix package is inefficient and unstable. A new package called YAMS recognizes dense blocks in MME, rearranges computations accordingly, allowing also for parallel computations for large dense blocks. With YAMS, GREML was much faster and more reliable.

Additional issues

This paper focused on algorithmic issues relevant to current mixed model computations in genomics in animal breeding. Many other issues exist and many will become relevant. For example, genotyping of animals with small amount of information leads to greater importance of avoidance of double counting in multi-step evaluations, or of more accurate match between genomic and additive relationships in ssGBLUP. With efforts to use sequence data, important operations are aligning short fragments, imputation, genomewide association studies, and incorporations of the detected variants in the evaluation. In dairy, a *de facto* global breeding scheme for Holsteins creates a need for a comprehensive genomic evaluation by Interbull that has a high accuracy, yet does not compromise privacy of the member countries. In general, while brute-force approaches may be formidable, as or more accurate yet inexpensive approaches may exist that exploit peculiarities of data including a limited effective population in farm populations. For instance, the sequence includes three G base pairs, but farm populations can be described by about 4000 to 20 000 chromosomal segments (Pocrnic *et al*., 2016b). This means a block size of about 150 to 750 kbases with hard to differentiate SNPs, and ability to reduce costs by working with blocks rather than individual SNP. This also suggests that increases of genomic accuracies past 4000 to 20 000 of high accuracy genotyped animals are limited (ignoring G × E and decays of predictivity over time). Similarly, if Interbull countries are unwilling to share genotypes on individual bulls or SNP effects derived from their population, they can contribute GEBV and genotypes of artificially generated animals (Maantysaari, 2015, personal communication) with their number and accuracies relevant to each country.

Conclusions

In conclusion, common operations in genetic selection are computationally feasible for an almost unlimited number of genotyped individuals using appropriate algorithms. Solving 'SNP' equations is best accomplished by iteration on data using either a special form of GS iteration or the PCG algorithm. Solving ssGBLUP is best accomplished by iteration on data using the PCG algorithm. For GREML, a sparse matrix package needs to account for dense blocks. Computations in the two latter cases greatly benefit from efficient (and sparse) inverse of the GRM.

Acknowledgments

Helpful suggestions from the two anonymous reviewers are gratefully acknowledged. This research was primarily supported by grants from Holstein Association USA (Brattleboro, VT), American Angus Association, Zoetis, Cobb-Vantress, Smithfield Premium Genetics, Pig Improvement Company, and the US Department of Agriculture's National Institute of Food and Agriculture (Agriculture

https://doi.org/10.1017/S1751731116002366 Published online by Cambridge University Press

and Food Research Initiative competitive grant 2015-67015-22936). The authors thank Ivan Pocrnic for help with editing.

References

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

Aguilar I, Misztal I, Johnson DL, Legarra A, Tsuruta S and Lawlor TJ 2010. A unified approach to utilize phenotypic, full pedigree, and genomic information for genetic evaluation of Holstein final score. Journal of Dairy Science 93, 743–752.

Aguilar I, Misztal I, Legarra A and Tsuruta S 2011. Efficient computation of genomic relationship matrix and other matrices used in single-step evaluation. Journal of Animal Breeding and Genetics 128, 422–428.

Christensen OF and Lund MS 2010. Genomic prediction when some animals are not genotyped. Genetics Selection Evolution 42, 2.

Colleau JJ 2002. An indirect approach to the extensive calculation of relationship coefficients. Genetics Selection Evolution 34, 409–421.

Faux P and Gengler N 2013. Inversion of a part of the numerator relationship matrix using pedigree information. Genetics Selection Evolution 45, 45.

Fernando RL, Dekkers JCM and Garrick DJ 2014. A class of Bayesian methods to combine large numbers of genotyped and non-genotyped animals for whole-genome analyses. Genetics Selection Evolution 46, 50.

Fragomeni BO, Lourenco DAL, Tsuruta S, Masuda Y, Aguilar I, Legarra A, Lawlor TJ and Misztal I 2015. Hot topic: use of genomic recursions in single-step genomic best linear unbiased predictor (BLUP) with a large number of genotypes. Journal of Dairy Science 98, 4090–4094.

George A and Liu JW 1981. Computer solution of large sparse positive definite systems. Prentice-Hall Inc, Englewood Hills, NJ, USA.

Gianola D, de los Campos G, Hill WG, Manfredi E and Fernando RL 2009. Additive genetic variability and the Bayesian alphabet. Genetics 183, 347–363.

Golden BL., Fernando RL and Garrick DJ 2016. Bolt and an alternative approach to genomic EPDs. Proceedings of the Beef Improvement Federation, June 14–17, 2016, Manhattan, KS, USA, pp. 102–106.

Legarra A, Aguilar I and Misztal I 2009. A relationship matrix including full pedigree and genomic information. Journal of Dairy Science 92, 4656–4663.

Legarra A, Christensen OF, Aguilar I and Misztal I 2014. Single step, a general approach for genomic selection. Livestock Science 166, 54–65.

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

Legarra A and Misztal I 2008. Computing strategies in genome-wide selection. Journal of Dairy Science 91, 360–366.

Liu Z, Goddard ME, Reinhardt F and Reents R 2014. A single-step genomic model with direct estimation of marker effects. Journal of Dairy Science 97, 5833–5850.

Lourenco DAL, Tsuruta S, Fragomeni BO, Masuda Y, Aguilar I, Legarra A, Bertrand JK, Amen TS, Wang L, Moser DW and Misztal I 2015. Genetic evaluation using single-step genomic BLUP in American Angus. Journal of Animal Science 93, 2653–2662.

Masuda Y, Misztal I, Tsuruta S, Legarra A, Aguilar I, Lourenco DAL, Fragomeni B and Lawlor TL 2016. Implementation of genomic recursions in single-step genomic BLUP for US Holsteins with a large number of genotyped animals. Journal of Dairy Science 99, 1968–1974.

Masuda Y, Tsuruta S, Aguilar I and Misztal I 2015. Technical note: acceleration of sparse operations for average-information REML analyses with supernodal methods and sparse-storage refinements. Journal of Animal Science 93, 4670–4674.

Meuwissen THE, Hayes BJ and Goddard ME 2001. Prediction of total genetic value using genome-wide dense marker maps. Genetics 157, 1819–1829.

Meuwissen THE, Svendsen M, Solberg T and Ødegård J 2015. Genomic predictions based on animal models using genotype imputation on a national scale in Norwegian Red cattle. Genetics Selection Evolution 47, 79.

Misztal I 2014. Computational techniques in animal breeding. Retrieved on 20 April 2016 from nce.ads.uga.edu.

Misztal I 2016. Inexpensive computation of the inverse of the genomic relationship matrix in populations with small effective population size. Genetics 202, 401–409.

Misztal I and Gianola D 1987. Indirect solution of mixed model equations. Journal of Dairy Science 70, 716–723.

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

Misztal I, Legarra A and Aguilar I 2014. Using recursion to compute the inverse of the genomic relationship matrix. Journal of Dairy Science 97, 3943–3952.

Misztal I and Perez-Enciso M 1993. Sparse matrix inversion for restricted maximum likelihood estimation of variance components by expectation-maximization. Journal of Dairy Science 76, 1479 –1483.

Mrode RA 2014. Linear models for the prediction of animal breeding values, 2nd and 3rd edition. CABI Publishing, Wallingford, UK.

Pérez-Enciso M, Misztal I and Elzo MA 1994. FSPAK: an interface for public domain sparse matrix subroutines. Proceedings of 5th World Congress on Genetics Applied to Livestock Production, 7–12 August, Guelph, ON, Canada 22, 87–88.

Pocrnic I, Lourenco DAL, Masuda Y, Legarra A and Misztal I 2016a. The dimensionality of genomic information and ts effect on genomic prediction. Genetics 203, 573–581.

Pocrnic I, Lourenco DAL, Masuda Y and Misztal I 2016b. Dimensionality of genomic information and performance of Algorithm for Proven and Young for different livestock species. Genetics Selection Evolution 48, 82.

Schaeffer LR 2006. Strategy for applying genome-wide selection in dairy cattle. Journal of Animal Breeding and Genetics 123, 218–223.

Schaeffer LR and Kennedy BW 1986. Computing strategies for solving mixed model equations. Journal of Dairy Science 69, 575–579.

Strandén I and Garrick DJ 2009. Technical note: derivation of equivalent computing algorithms for genomic predictions and reliabilities of animal merit. Journal of Dairy Science 92, 2971–2975.

Strandén I and Christensen OF 2011. Allele coding in genomic evaluation. Genetics Selection Evolution 43, 25.

Strandén I and Lidauer M 1999. Solving large mixed linear models using preconditioned conjugate gradient iteration. Journal of Dairy Science 82, 2779–2787.

Strandén I and Mäntysaari EA 2014. Comparison of some equivalent equations to solve single-step GBLUP. In Proceedings of 10th World Congress on Genetics Applied to Livestock Production, 17–22 August, Vancouver, BC, Canada.

Sun X, Qu L, Garrick DJ, Dekkers JCM and Fernando RL 2012. A fast EM algorithm for BayesA-like prediction of genomic breeding values. PLoS One 7, e49157.

Takahashi K, Fagan J and Chen MS 1973. Formation of a sparse bus impedance matrix and its application to short circuit study. In Proceedings of 8th Power Industry Computer Applications Conference, 3–6 June, Minneapolis, MN, USA, p. 63.

Tsuruta S, Misztal I and Stranden I 2001. Use of the preconditioned conjugate gradient algorithm as a generic solver for mixed-model equations in animal breeding applications. Journal of Animal Science 79, 1166–1172.

VanRaden PM 2008. Efficient methods to compute genomic predictions. Journal of Dairy Science 91, 4414–4423.

VanRaden PM, Van Tassell CP, Wiggans GR, Sonstegard TS, Schnabel RD, Taylor JF and Schenkel FS 2009. Invited review: reliability of genomic predictions for North American Holstein bulls. Journal of Dairy Science 92, 16–24.

Wang H, Misztal I, Aguilar I, Legarra A and Muir WM 2012. Genome-wide association mapping including phenotypes from relatives without genotypes. Genetics Research 94, 73–83.

Wang H, Misztal I and Legarra A 2014. Differences between genomic-based and pedigree-based relationships in a chicken population, as a function of quality control and pedigree links among individuals. Journal of Animal Breeding and Genetics 131, 445–451.

Zhang Z, Liu J, Ding X, Bijma P, de Koning DJ and Zhang Q 2010. Best linear unbiased prediction of genomic breeding values using a trait-specific marker-derived relationship matrix. PLoS One 5, e12648.

https://doi.org/10.1017/S1751731116002366 Published online by Cambridge University Press