frontiers in Genetics

METHODS
published: 25 June 2019
doi: 10.3389/fgene.2019.00596

An Efficient Method to Calculate Genomic Prediction Accuracy for New Individuals

Mohammad H. Ferdosi, Natalie K. Connors and Bruce Tier*

Animal Genetics and Breeding Unit, University of New England, Armidale, NSW, Australia

Diagonal elements of the coefficient matrix are necessary to calculate the genomic prediction accuracy. Here an improved methodology is described, to update the inverse of the coefficient matrix (**C**) for new individuals with a genotype, with and without phenotypes. Computational performance is significantly improved by re-using parts of the coefficient matrix inverse calculations that do not change from one animal to another, in combination with updated calculations for those that do change. This method expedites calculation of accuracy for new individuals with genotypes, without re-doing the whole population, by using the previously calculated matrices.

Keywords: accuracy, efficient, breeding value, prediction, method

OPEN ACCESS

Edited by:
Guilherme J. M. Rosa,
University of Wisconsin-Madison,
United States

Reviewed by:
Breno De Oliveira Fragomeni,
University of Connecticut,
United States
Nicolas Gengler,
Gembloux Agro-Bio Tech, University
of Liege, Belgium

****Correspondence:***
Mohammad H. Ferdosi
mferdos3@une.edu.edu

Specialty section:
This article was submitted to
Livestock Genomics,
a section of the journal
Frontiers in Genetics

Received: *10 January 2019*
Accepted: *05 June 2019*
Published: *25 June 2019*

Citation:
Ferdosi MH, Connors NK and Tier B
(2019) An Efficient Method to
Calculate Genomic Prediction
Accuracy for New Individuals.
Front. Genet. 10:596.
doi: 10.3389/fgene.2019.00596

1. INTRODUCTION

In the last decade, technological advances have significantly decreased genotyping costs, particularly for agricultural livestock and cropping species. This reduction in costs has enabled regular genomic best linear unbiased prediction (GBLUP) (VanRaden, 2008) analyses for the production of genomic breeding values. Currently, genomic information is routinely used in the Australian beef industry in producing estimated breeding values (EBVs). Low costs and high industry uptake has resulted in a rapid increase in the number of new genotypes and thus the size of the genomic population is growing larger. GBLUP requires inversion of the genomic relationship matrix (**G**) and the coefficient matrix (**C**), which is computationally demanding. More efficient methods such as APY (Algorithm for Proven and Young animals) (Misztal, 2016) and PICD (Partial Incomplete Cholesky Decomposition) (Hancock, 2017) can handle large numbers of genotyped animals by approximating the inverse of **G** only and not the coefficient matrix. However, these approaches do not address the need for diagonal elements of the coefficient matrix inverse (left hand side) required for calculating EBV accuracies. With the increasing speed at which new genotypes are provided, inversion of the coefficient matrix for accuracy calculation is increasingly computationally demanding and time consuming, requiring more efficient methods.

Here we propose a method to calculate the accuracy of new individuals, with and without phenotypes, by updating the coefficient matrix inverse ($\mathbf{C}^{-1}$) for new individuals only, without re-doing the whole population. Using this method, we significantly reduce time and computational demand by updating the accuracy of new individuals and reducing redundancy in the reference population.

2. METHODS

2.1. Theory

We consider a simple animal model without fixed effects. This model is

$$\mathbf{y} = \mathbf{Z}\mathbf{u} + \mathbf{e} \tag{1}$$

where $\mathbf{y}$, $\mathbf{Z}$, $\mathbf{u}$, and $\mathbf{e}$ are vector of observations, design matrix, vector of solutions, and vector of random residual effects, respectively. The solutions and residual variances are $var(\mathbf{u}) = \mathbf{G}\sigma_u^2$ and $var(\mathbf{e}) = \mathbf{I}\sigma_e^2$. The mixed model equations (MME) for the above model are

$$\mathbf{C}\mathbf{u} = [\mathbf{Z}'\mathbf{Z} + \alpha\mathbf{G}^{-1}]\mathbf{u} = \mathbf{r}, \tag{2}$$

where $\mathbf{C}$ is the coefficient matrix and $\alpha = \frac{\sigma_e^2}{\sigma_u^2}$. Henderson derived a method by using the diagonal of $\mathbf{C}^{-1}$ and the diagonal of $\mathbf{G}$ to calculate the accuracy of each estimated breeding value (Henderson, 1975). Accordingly, the accuracy can be calculated with this formula $\sqrt{1 - \alpha\frac{\mathrm{c^{ii}}}{\mathrm{g_{ii}}}}$, where $\mathrm{c^{ii}}$ is the diagonal element of $\mathbf{C}^{-1}$ for individual i, and $\mathrm{g_{ii}}$ is the diagonal element of $\mathbf{G}$ for individual i.

2.1.1. Updating C^{-1} to Illustrate the Proposed Method

To calculate the accuracy of individuals with or without phenotypes, each individual can be added to $\mathbf{C}^{-1}$ separately. In this case, the partitioned matrix of MME (Equation 2) is

$$\begin{bmatrix} \mathbf{C}_{\mathrm{pp}} & \mathbf{C}_{\mathrm{pq}} \\ \mathbf{C}'_{\mathrm{pq}} & \mathbf{C}_{\mathrm{qq}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathrm{p}} \\ \mathbf{u}_{\mathrm{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{\mathrm{p}} \\ \mathbf{r}_{\mathrm{q}} \end{bmatrix} \tag{3}$$

where subscript p and q are core individuals forming the reference population and new individuals respectively. New individuals may or may not have phenotypes.

As demonstrated in Equation (2), $\mathbf{Z}'\mathbf{Z}$ becomes $\begin{pmatrix} (\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}} & 0 \\ 0 & (\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}} \end{pmatrix}$ where $(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}$ is a diagonal matrix with dimension equal to the number of core animals in the population, and the $(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}$ in the lower diagonal represents new animals. Since $\mathbf{G}^{-1}$ becomes $\begin{pmatrix} \mathbf{G}_{\mathrm{pp}} & \mathbf{G}_{\mathrm{pq}} \\ \mathbf{G}'_{\mathrm{pq}} & \mathbf{G}_{\mathrm{qq}} \end{pmatrix}^{-1}$, and $\mathbf{C}^{-1}$ becomes

$$[\mathbf{Z}'\mathbf{Z} + \alpha\mathbf{G}^{-1}]^{-1} \approx \left(\begin{bmatrix} (\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}} & 0 \\ 0 & (\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}} \end{bmatrix} + \alpha \begin{bmatrix} \mathbf{G}_{\mathrm{pp}} & \mathbf{G}_{\mathrm{pq}} \\ \mathbf{G}'_{\mathrm{pq}} & \mathbf{G}_{\mathrm{qq}} \end{bmatrix}^{-1} \right)^{-1} \tag{4}$$

based on Equations (2) and (3). Inverting $\mathbf{C}$ is computationally demanding, as both $\mathbf{G}$ and the entire $\mathbf{C}$ should be inverted in each analysis for all individuals. $\mathbf{G}_{\mathrm{pp}}^{-1}$ needs to be updated as the new individuals are added to $\mathbf{G}$. This can be performed by the method explained in Meyer et al. (2013). However, since we want to know the accuracy, we must invert $\mathbf{C}$ as well as $\mathbf{G}$. Equation (4) can be converted with the following inversion lemma which is equivalent to the Woodbury's formula (Henderson, 1963, 1975; Henderson and Searle, 1981):

$$(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{E})^{-1} = \boldsymbol{E}^{-1}\boldsymbol{D}(\boldsymbol{D} - \boldsymbol{E}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1}\boldsymbol{E}\boldsymbol{A}^{-1}. \tag{5}$$

With $\mathbf{C}$ we can consider $\boldsymbol{A} = \mathbf{Z}'\mathbf{Z}$, $\boldsymbol{B} = -\mathbf{I}$, $\boldsymbol{D}^{-1} = \mathbf{G}^{-1}$ and $\boldsymbol{E} = \alpha\mathbf{I}$. Thus $\mathbf{C}^{-1}$ is

$$\begin{aligned} (\mathbf{Z}'\mathbf{Z} + \alpha\mathbf{G}^{-1})^{-1} &= \alpha^{-1}\mathbf{G}(\mathbf{G} + \alpha(\mathbf{Z}'\mathbf{Z})^{-1})^{-1}\alpha(\mathbf{Z}'\mathbf{Z})^{-1} \\ &= \mathbf{G}\mathbf{M}^{-1}(\mathbf{Z}'\mathbf{Z})^{-1} \end{aligned} \tag{6}$$

where $\mathbf{M}^{-1}$ as $(\mathbf{G} + \alpha(\mathbf{Z}'\mathbf{Z})^{-1})^{-1}$ is used for simplification and is shown below in Equation (8). For the partitioned matrices in Equation (4), $\mathbf{C}^{-1}$ becomes

$$\begin{aligned} &\left(\begin{bmatrix} (\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}} & 0 \\ 0 & (\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}} \end{bmatrix} + \alpha \begin{bmatrix} \mathbf{G}_{\mathrm{pp}} & \mathbf{G}_{\mathrm{pq}} \\ \mathbf{G}'_{\mathrm{pq}} & \mathbf{G}_{\mathrm{qq}} \end{bmatrix}^{-1} \right)^{-1} \\ &\approx \begin{bmatrix} \mathbf{G}_{\mathrm{pp}} & \mathbf{G}_{\mathrm{pq}} \\ \mathbf{G}'_{\mathrm{pq}} & \mathbf{G}_{\mathrm{qq}} \end{bmatrix} \left(\begin{bmatrix} \mathbf{G}_{\mathrm{pp}} & \mathbf{G}_{\mathrm{pq}} \\ \mathbf{G}'_{\mathrm{pq}} & \mathbf{G}_{\mathrm{qq}} \end{bmatrix} + \alpha \begin{bmatrix} (\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1} & 0 \\ 0 & (\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} \end{bmatrix} \right)^{-1} \\ &\qquad \begin{bmatrix} (\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1} & 0 \\ 0 & (\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} \end{bmatrix} \\ &\approx \begin{bmatrix} \mathbf{G}_{\mathrm{pp}} & \mathbf{G}_{\mathrm{pq}} \\ \mathbf{G}'_{\mathrm{pq}} & \mathbf{G}_{\mathrm{qq}} \end{bmatrix} \begin{bmatrix} \mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1} & \mathbf{G}_{\mathrm{pq}} \\ \mathbf{G}'_{\mathrm{pq}} & \mathbf{G}_{\mathrm{qq}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} \end{bmatrix}^{-1} \\ &\qquad \begin{bmatrix} (\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1} & 0 \\ 0 & (\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} \end{bmatrix}, \end{aligned} \tag{7}$$

where $\approx$ is approximation sign. By using lemma (6) $\mathbf{G}^{-1}$ is not required and we only need to invert the middle matrix ($\mathbf{M}$) in Equation (7). With this simplification $\mathbf{M}^{-1}$ can be updated for each new individual using Cholesky decomposition and multiplying the Cholesky factors, i.e., $\mathbf{M}^{-1} = \mathbf{L}^{-T}\mathbf{L}^{-1}$ (Harville, 1997; Meyer et al., 2013).

$$\begin{aligned} \mathbf{M}^{-1} &= \begin{bmatrix} \mathbf{L}_{\mathrm{pp}}^{-T}\mathbf{L}_{\mathrm{pp}}^{-1} + \mathbf{L}_{\mathrm{pp}}^{-T}\mathbf{L}'_{\mathrm{qp}}\mathbf{L}_{\mathrm{qq}}^{-T}\mathbf{L}_{\mathrm{qq}}^{-1}\mathbf{L}_{\mathrm{qp}}\mathbf{L}_{\mathrm{pp}}^{-1} & -\mathbf{L}_{\mathrm{pp}}^{-T}\mathbf{L}'_{\mathrm{pq}}\mathbf{L}_{\mathrm{qq}}^{-T}\mathbf{L}_{\mathrm{qq}}^{-1} \\ -\mathbf{L}_{\mathrm{qq}}^{-T}\mathbf{L}_{\mathrm{qq}}^{-1}\mathbf{L}_{\mathrm{qp}}\mathbf{L}_{\mathrm{pp}}^{-1} & \mathbf{L}_{\mathrm{qq}}^{-T}\mathbf{L}_{\mathrm{qq}}^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 \\ \mathbf{S}_3 & \mathbf{S}_4 \end{bmatrix}. \end{aligned} \tag{8}$$

Therefore, Equation (7) can be written as

$$\begin{aligned} \begin{bmatrix} \mathbf{C}_{\mathrm{pp}} & \mathbf{C}_{\mathrm{pq}} \\ \mathbf{C}'_{\mathrm{pq}} & \mathbf{C}_{\mathrm{qq}} \end{bmatrix}^{-1} &= \begin{bmatrix} \mathbf{G}_{\mathrm{pp}} & \mathbf{G}_{\mathrm{pq}} \\ \mathbf{G}'_{\mathrm{pq}} & \mathbf{G}_{\mathrm{qq}} \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 \\ \mathbf{S}_3 & \mathbf{S}_4 \end{bmatrix} \begin{bmatrix} (\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1} & 0 \\ 0 & (\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{G}_{\mathrm{pp}}\mathbf{S}_1 + \mathbf{G}_{\mathrm{pq}}\mathbf{S}_3 & \mathbf{G}_{\mathrm{pp}}\mathbf{S}_2 + \mathbf{G}_{\mathrm{pq}}\mathbf{S}_4 \\ \mathbf{G}'_{\mathrm{pq}}\mathbf{S}_1 + \mathbf{G}_{\mathrm{qq}}\mathbf{S}_3 & \mathbf{G}'_{\mathrm{pq}}\mathbf{S}_2 + \mathbf{G}_{\mathrm{qq}}\mathbf{S}_4 \end{bmatrix} \begin{bmatrix} (\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1} & 0 \\ 0 & (\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} \end{bmatrix} \\ &= \begin{bmatrix} (\mathbf{G}_{\mathrm{pp}}\mathbf{S}_1 + \mathbf{G}_{\mathrm{pq}}\mathbf{S}_3)(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1} & (\mathbf{G}_{\mathrm{pp}}\mathbf{S}_2 + \mathbf{G}_{\mathrm{pq}}\mathbf{S}_4)(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} \\ (\mathbf{G}'_{\mathrm{pq}}\mathbf{S}_1 + \mathbf{G}_{\mathrm{qq}}\mathbf{S}_3)(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1} & (\mathbf{G}'_{\mathrm{pq}}\mathbf{S}_2 + \mathbf{G}_{\mathrm{qq}}\mathbf{S}_4)(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} \end{bmatrix}, \end{aligned} \tag{9}$$

based on Equation (8) $\mathbf{S}_2 = -\mathbf{L}_{\mathrm{pp}}^{-T}\mathbf{L}'_{\mathrm{qp}}\mathbf{L}_{\mathrm{qq}}^{-T}\mathbf{L}_{\mathrm{qq}}^{-1}$ and $\mathbf{S}_4 = \mathbf{L}_{\mathrm{qq}}^{-T}\mathbf{L}_{\mathrm{qq}}^{-1}$. By multiplying back the Cholesky factors of $\mathbf{M}$ the solutions for $\mathbf{L}'_{\mathrm{pq}}$ and $\mathbf{L}_{\mathrm{qq}}\mathbf{L}_{\mathrm{qq}}^{T}$ are

$$\mathbf{L}'_{\mathrm{qp}} = \mathbf{L}_{\mathrm{pp}}^{-1}\mathbf{G}_{\mathrm{pq}} \tag{10}$$

and

$$\mathbf{L}_{\mathrm{qq}}\mathbf{L}_{\mathrm{qq}}^{T} = \mathbf{G}_{\mathrm{qq}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} - \mathbf{G}'_{\mathrm{pq}}(\mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1})^{-1}\mathbf{G}_{\mathrm{pq}}, \tag{11}$$

and $\mathbf{C}^{\mathrm{qq}}$ which is the inverse of $\mathbf{C}_{\mathrm{qq}}$ becomes

$$\begin{aligned}\mathbf{C}^{\mathrm{qq}} &= (\mathbf{G}'_{\mathrm{pq}}(-\mathbf{L}_{\mathrm{pp}}^{-T}\mathbf{L}'_{\mathrm{qp}}\mathbf{L}_{\mathrm{qq}}^{-T}\mathbf{L}_{\mathrm{qq}}^{-1}) + \mathbf{G}_{\mathrm{qq}}(\mathbf{L}_{\mathrm{qq}}^{-T}\mathbf{L}_{\mathrm{qq}}^{-1}))(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} \\ &= (\mathbf{G}'_{\mathrm{pq}}(-\mathbf{L}_{\mathrm{pp}}^{-T}\mathbf{L}'_{\mathrm{pq}}\mathbf{S}_4) + \mathbf{G}_{\mathrm{qq}}\mathbf{S}_4)(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} \\ &= \mathbf{S}_4(\mathbf{G}'_{\mathrm{pq}}(-\mathbf{L}_{\mathrm{pp}}^{-T}\mathbf{L}'_{\mathrm{qp}}) + \mathbf{G}_{\mathrm{qq}})(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} \\ &= \frac{-\mathbf{G}'_{\mathrm{pq}}\mathbf{L}_{\mathrm{pp}}^{-T}\mathbf{L}'_{\mathrm{qp}} + \mathbf{G}_{\mathrm{qq}}}{(\mathbf{G}_{\mathrm{qq}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} - \mathbf{G}'_{\mathrm{pq}}(\mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1})^{-1}\mathbf{G}_{\mathrm{pq}})(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}} \\ &= \frac{-\mathbf{G}'_{\mathrm{pq}}\mathbf{L}_{\mathrm{pp}}^{-T}\mathbf{L}_{\mathrm{pp}}^{-1}\mathbf{G}_{\mathrm{pq}} + \mathbf{G}_{\mathrm{qq}}}{(\mathbf{G}_{\mathrm{qq}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} - \mathbf{G}'_{\mathrm{pq}}(\mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1})^{-1}\mathbf{G}_{\mathrm{pq}})(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}}\end{aligned} \tag{12}$$

$$\mathbf{C}^{\mathrm{qq}} = \frac{-\mathbf{G}'_{\mathrm{pq}}\mathbf{L}_{\mathrm{pp}}^{-T}\mathbf{L}_{\mathrm{pp}}^{-1}\mathbf{G}_{\mathrm{pq}} + \mathbf{G}_{\mathrm{qq}}}{(\mathbf{G}_{\mathrm{qq}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} - \mathbf{G}'_{\mathrm{pq}}(\mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1})^{-1}\mathbf{G}_{\mathrm{pq}})(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}} \tag{13}$$

where "—" is the right division sign (multiplying numerator by inverse of denominator) and $\mathbf{L}_{\mathrm{pp}}^{-T}\mathbf{L}_{\mathrm{pp}}^{-1} = (\mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1})^{-1}$ so

$$\mathbf{C}^{\mathrm{qq}} = \frac{-\mathbf{G}'_{\mathrm{pq}}(\mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1})^{-1}\mathbf{G}_{\mathrm{pq}} + \mathbf{G}_{\mathrm{qq}}}{(\mathbf{G}_{\mathrm{qq}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} - \mathbf{G}'_{\mathrm{pq}}(\mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1})^{-1}\mathbf{G}_{\mathrm{pq}})(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}} \tag{14}$$

Only $\mathbf{G}'_{\mathrm{pq}}$, $\mathbf{G}_{\mathrm{pq}}$, $\mathbf{G}_{\mathrm{qq}}$, and $(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}$ change with each new genotyped individual.

For animals without phenotypes $(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}$ is a null matrix and the denominator in equation 14 becomes zero. However, we can assume the limit approach to α as $(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}$ approach to zero. Thus, $\mathbf{C}^{\mathrm{qq}}$ is

$$\begin{aligned}&\lim_{(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}\to 0} \frac{-\mathbf{G}'_{\mathrm{pq}}(\mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1})^{-1}\mathbf{G}_{\mathrm{pq}} + \mathbf{G}_{\mathrm{qq}}}{(\mathbf{G}_{\mathrm{qq}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} - \mathbf{G}'_{\mathrm{pq}}(\mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1})^{-1}\mathbf{G}_{\mathrm{pq}})(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}} \\ &\qquad = \frac{-\mathbf{G}'_{\mathrm{pq}}(\mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1})^{-1}\mathbf{G}_{\mathrm{pq}} + \mathbf{G}_{\mathrm{qq}}}{\alpha}\end{aligned} \tag{15}$$

In summary, Equations (14) and (15) can be used to calculate the prediction accuracies of individuals with and without phenotype, respectively.

2.1.2. Updating the $\mathbf{M}^{-1}$ for New Individuals

Based on Equations (14) and (15) only $(\mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1})^{-1} = \mathbf{M}_{\mathrm{pp}}^{-1}$ changes (see Equation 7) in order to update the reference population. The updated $\mathbf{M}_{\mathrm{pp}}^{-1}$, i.e., $\mathbf{M}_{\mathrm{pp}_{new}}^{-1}$ is

$$\begin{aligned}\mathbf{M}_{\mathrm{pp}_{new}}^{-1} &= \begin{bmatrix}\mathbf{S}_{n1} & \mathbf{S}_{n2} \\ \mathbf{S}_{n3} & \mathbf{S}_{n4}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{M}^{-1} - \mathbf{S}_2\mathbf{G}'_{\mathrm{pq}}\mathbf{M}^{-1} & -\mathbf{M}^{-1}\mathbf{G}_{\mathrm{pq}}\mathbf{S}_4 \\ \mathbf{S}'_{n2} & (\mathbf{G}_{\mathrm{qq}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{qq}}^{-1} - \mathbf{G}'_{\mathrm{pq}}\mathbf{M}^{-1}\mathbf{G}_{\mathrm{pq}})^{-1}\end{bmatrix}\end{aligned} \tag{16}$$

by regarding previous assumptions and Equation (8). $\mathbf{M}^{-1}$ is the largest matrix that was generated in the previous run and can be compressed and stored in binary format to avoid memory issues. The other matrices were small and can be built efficiently by using optimized Linear Algebra PACKage (LAPACK). The equations (14, 15, and 16) were implemented as an R function (**Appendix A**) to show the prototype and in C++ with Armadillo library (Sanderson and Curtin, 2016) to assess its performance—single thread.

2.1.3. Simulated Data

Matrices with seven columns representing seven single-nucleotide polymorphisms for each individual and 1000, 2000, 3000, ... 24000, and 25000 rows were created and filled with 0 (AA), 1 (AB), and 2 (BB) randomly. The genomic relationship matrices (**G**) were built by using VanRaden (2008) method 1, with dimension individuals by individuals. Importantly, increasing the number to SNPs does not affect the computational time. These matrices were used to assess the performance of the proposed method to calculate accuracy.

2.1.4. Performance Evaluation

To evaluate performance, each set was run in three steps. In the first step, the elapsed time to build the coefficient matrix by using the classic approach (i.e., inverting **G** and **C**) was measured. In the second step, the time to build $(\mathbf{G}_{\mathrm{pp}} + \alpha(\mathbf{Z}'\mathbf{Z})_{\mathrm{pp}}^{-1})^{-1}$—initial matrices required to update c^{qq} was measured. In the third step, the time to calculate c^{qq} by using the initial matrices was measured.

3. RESULTS AND DISCUSSIONS

By calculating the accuracy of young individuals using Equations (14) and (15) computational times have been significantly reduced. Computational performance using this method is considerably faster, in comparison with existing methods, as shown in **Figure 1**, with only negligible differences in accuracy due to rounding errors (less than 8.88×10^{-16}). The proposed approach using $\mathbf{Z}'\mathbf{Z}$ (a diagonal matrix) resulted in shorter time to build the matrices used to update c^{qq} compared to when using the classic approach to calculate accuracies. This method can be extended in order to accommodate fixed effects and dense $\mathbf{Z}'\mathbf{Z}$ when c^{qq} is updated. Furthermore, the part of **C** for animals with phenotype ($\mathbf{C}^{\mathrm{pp}}$) must be updated as more individuals are phenotyped.

This method could be exploited within routine breeding value estimation for expidited accuracy calculations. Breeding value accuracy is based on an individual's relatedness to the core reference, such that high accuracy indicates high relatedness. This method to calculate accuracy will affect how the genotypes are used, based on how informative they are for the prediction, improving efficiency by reducing redundant information.

New individuals with phenotypes and low accuracy can be added to the core population, as it is likely these animals are lowly related. Their addition improves the diversity and informativity of the core reference population, and can further improve imputation accuracy of the missing genotypes, with added

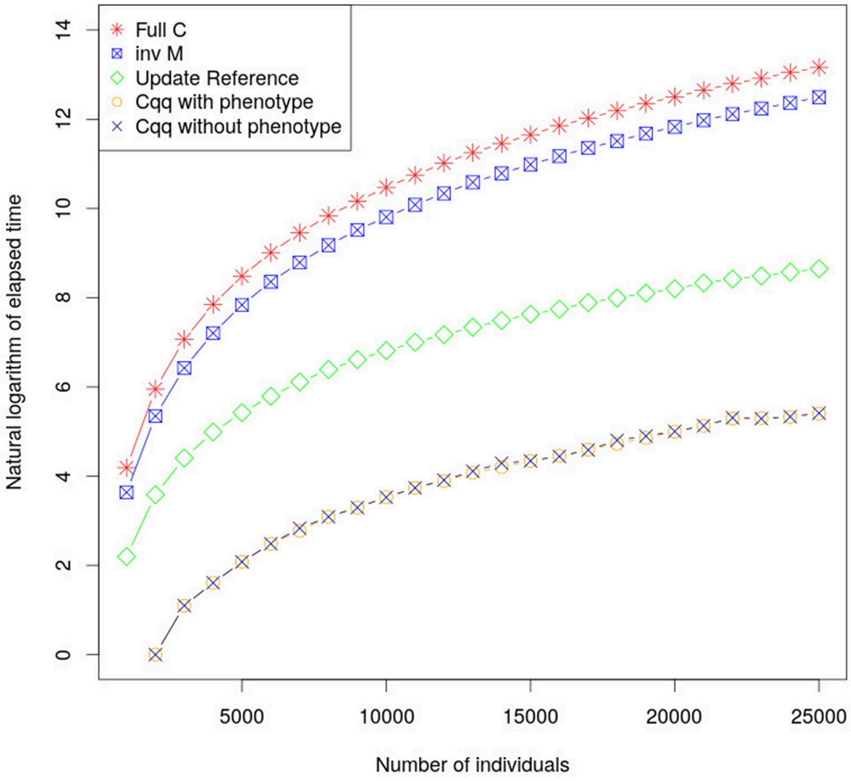

FIGURE 1 | The graph shows the elapsed time required to calculate c^{qq} using different approaches. **inv M** is the elapsed time to calculate the **M** matrix which is the core element to calculate c^{qq} for a new individual. **Update Reference** is the elapse time to update **M** matrix, the $\mathbf{c^{qq}}$ **with and without phenotype** shows the elapsed time to calculate c^{qq} when there is or there is not phenotype for the individual, respectively. Their performances were very similar, and as such the lines overlap.

diversity into the imputation haplotype library. Individuals with high accuracy are not required to be added to the core, with or without phenotype, as their accuracy indicates their relatives are already included in this reference population, making their addition redundant. New individuals without phenotypes and low accuracy, should have relatives genotyped to improve accuracy and/or should have their phenotypes recorded to improve the core population.

It is possible to exploit the accuracy calculation as a type of quality control filter for population data, such that individuals with an expected level of relatedness to the reference population, obtains a low accuracy, this may be indicative of genotyping/sampling error, mis-assigned breed, etc. The rapid accuracy calculation for those individuals without phenotype can provide important context for quickly developing a phenotyping strategy.

4. CONCLUSION

Updating the inverse of **C** for new individuals with and without phenotype, using the method here, is shown to reduce the computational effort significantly. With increasing numbers of genotyped animals in genetic evaluations, computational efficiency is essential for frequent and timely evaluations. This method provides an improved and efficient method to deliver accurate and fast evaluations when few new young individuals are genotyped but may or may not have phenotypes.

AUTHOR CONTRIBUTIONS

MF developed the method, structured the manuscript, and wrote the method and theory. NC wrote the introduction, result and discussion, and performed major revision. BT gave some comments to improve the final method and article.

FUNDING

MF, NC, and BT were supported by Meat and Livestock Australia project L.GEN.0174.

ACKNOWLEDGMENTS

Parts of this article has been published in proceeding of World Congress on Genetics Applied to Livestock Production (WCGALP)—(Ferdosi et al., 2018). The authors want to thank Daniela A. L. Lourenco and Karin Meyer for their comments on the WCGALP paper. The authors also wish to thank the reviewers for comments leading to the clarification and improvement of this article.

REFERENCES

Ferdosi, M., Connors, N., and Tier, B. (2018). "An efficient method to calculate accuracy of estimated breeding values for individuals without phenotypes," in *Proceedings of the World Congress on Genetics Applied to Livestock Production, Electronic Poster Session - Methods and Tools - Models and Computing Strategies* (Auckland).

Hancock, T. P. (2017). "Approximate gblup for efficient routine evaluations," in *Association for the Advancement of Animal Breeding and Genetics* (Townsville, QLD).

Harville, D. A. (1997). *Matrix Algebra From a Statistician's Perspective, Vol. 1.* Yorktown Heights, NY: Springer.

Henderson, C. R. (1963). Selection index and expected genetic advance. *Stat. Genet. Plant Breed.* 982, 141–163.

Henderson, C. R. (1975). Best linear unbiased estimation and prediction under a selection model. *Biometrics* 31, 423–447.

Henderson, H. V., and Searle, S. R. (1981). On deriving the inverse of a sum of matrices. *SIAM Rev.* 23, 53–60.

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

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

Sanderson, C., and Curtin, R. (2016). Armadillo: a template-based c++ library for linear algebra. *J. Open Source Softw*. 1:26. doi: 10.21105/joss.00026

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

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

APPENDIX A

R functions that show the prototype.

```
# Create a partition matrix from 4 matrices
bind <− function(A, B, C, D)
{
  rbind(cbind(A, B), cbind(C, D))
}

# Create a zero matrix
zM <− function(row, col)
{
  matrix(0, nrow = row, ncol = col)
}

# Calculate X'X
xpx <− function(x)
{
  t(x) %∗% x
}

# Calculate C inverse by using traditional method
tradMethod <− function(Zp, Zq, Gpp, Gpq, Gqq, Alpha)
{
  A <− xpx(Zp)
  D <− xpx(Zq)
  B <− zM(nrow(A), ncol(D))
  C <− t(B)
  solve(bind(A, B, C, D) + Alpha ∗
  solve(bind(Gpp, Gpq, t(Gpq), Gqq)))
}

# Calculate Cqq when there are phenotypes for new
# individuals
cqqPhen <− function(Zp, Zq, Gpp, Gpq, Gqq, Alpha)
{
  Minv <− solve(Gpp + Alpha ∗ solve(xpx(Zp)))
  n1 <− −t(Gpq) %∗% Minv %∗% Gpq + Gqq
  n2 <− (Gqq + Alpha ∗ solve(xpx(Zq)) − t(Gpq) %∗%
  Minv %∗% Gpq) %∗% xpx(Zq)
  n1 %∗% solve(n2)
}

# Calculate Cqq when there are no phenotypes for
new individuals
cqq <− function(Zp, Zq, Gpp, Gpq, Gqq, Alpha)
{
  Minv <− solve(Gpp + Alpha ∗ solve(xpx(Zp)))
  n1 <− −t(Gpq) %∗% Minv %∗% Gpq + Gqq
  n1 / Alpha
}
```