A computationally efficient algorithm to leverage average information REML for (co)variance component estimation in the genomic era

Ismo Strandén, Esa A. Mäntysaari, Martin H. Lidauer, Robin Thompson, Hongding Gao

Genetics Selection Evolution

RESEARCH ARTICLE

Open Access

A computationally efficient algorithm to leverage average information REML for (co)variance component estimation in the genomic era

Ismo Strandén[1], Esa A. Mäntysaari[1], Martin H. Lidauer[1], Robin Thompson[2] and Hongding Gao[1*]

Abstract

Background Methods for estimating variance components (VC) using restricted maximum likelihood (REML) typically require elements from the inverse of the coefficient matrix of the mixed model equations (MME). As genomic information becomes more prevalent, the coefficient matrix of the MME becomes denser, presenting a challenge for analyzing large datasets. Thus, computational algorithms based on iterative solving and Monte Carlo approximation of the inverse of the coefficient matrix become appealing. While the standard average information REML (AI-REML) is known for its rapid convergence, its computational intensity imposes limitations. In particular, the standard AI-REML requires solving the MME for each VC, which can be computationally demanding, especially when dealing with complex models with many VC. To bridge this gap, here we (1) present a computationally efficient and tractable algorithm, named the augmented AI-REML, which facilitates the AI-REML by solving an augmented MME only once within each REML iteration; and (2) implement this approach for VC estimation in a general framework of a multi-trait GBLUP model. VC estimation was investigated based on the number of VC in the model, including a two-trait, three-trait, four-trait, and five-trait GBLUP model. We compared the augmented AI-REML with the standard AI-REML in terms of computing time per REML iteration. Direct and iterative solving methods were used to assess the advances of the augmented AI-REML.

Results When using the direct solving method, the augmented AI-REML and the standard AI-REML required similar computing times for models with a small number of VC (the two- and three-trait GBLUP model), while the augmented AI-REML demonstrated more notable reductions in computing time as the number of VC in the model increased. When using the iterative solving method, the augmented AI-REML demonstrated substantial improvements in computational efficiency compared to the standard AI-REML. The elapsed time of each REML iteration was reduced by 75%, 84%, and 86% for the two-, three-, and four-trait GBLUP models, respectively.

Conclusions The augmented AI-REML can considerably reduce the computing time within each REML iteration, particularly when using an iterative solver. Our results demonstrate the potential of the augmented AI-REML as an appealing approach for large-scale VC estimation in the genomic era.

*Correspondence:
Hongding Gao
hongding.gao@luke.fi
[1] Natural Resources Institute Finland (Luke), 31600 Jokioinen, Finland
[2] Rothamsted Research, Harpenden, Herts AL5 2JQ, UK

Background

Accurate estimates of (co)variance components (VC) are crucial in computing precise genetic and genomic predictions. In the context of plant breeding, VC are typically estimated within each cycle of genomic prediction.

Conversely, in animal breeding, VC used for genomic prediction are often initially estimated using an animal model with pedigree information and subsequently updated at regular intervals. Although unbiased estimates of VC can be obtained using complete data and a pedigree-based model, results from earlier studies have shown that ignoring the genomic information for populations undergoing intense genomic selection yielded biased estimates of VC [1, 2]. Notably, results from the U.S. dairy cattle genomic evaluation showed that prediction bias decreased when heritability was reduced by about 50% to 70%, indicating an overestimated heritability [3, 4]. Given these findings, there is a growing consensus that the VC estimation needs to be done properly in the genomic era [5, 6]. Consequently, continued attention and refinement are essential for accurate VC estimation.

Restricted maximum likelihood (REML) serves as an important method in genetic analysis. It facilitates VC estimation within multivariate linear mixed models, which are commonly employed in the fields of animal and plant breeding to account for genetic correlations between traits and to make better use of available information across traits. Unlike maximum likelihood estimation, which estimates parameters using the likelihood function, REML adjusts the likelihood function using error contrasts. This adjustment accounts for the loss of degrees of freedom associated with estimating the fixed effects [7]. By using error contrasts, REML produces less biased estimates of VC, making it a widely applied approach in animal and plant breeding [5, 8–10].

Various approaches are available for maximizing the REML likelihood. The two most widely used methods are the expectation–maximization REML (EM-REML) and the average information REML (AI-REML) [11, 12]. EM-REML relies on the first derivatives of the REML log-likelihood, which make it straightforward to implement but suffers from slow convergence. In contrast, AI-REML uses both the first and second derivatives of the REML log-likelihood, resulting in significantly quicker convergence rates [13]. All Newton-type methods such as AI-REML use the vector of first derivatives at the current estimates, along with a matrix that characterizes the information content of the unknown VC in the analysis. The popularity of AI-REML is attributed to its quick convergence compared to the EM-REML; however, AI-REML is computationally more intensive [12, 13].

The analytical REML-based methods, typically used for VC estimation, require elements from the inverse coefficient matrix of the mixed model equations (MME), i.e., the prediction error (co)variances (PEV/PEC). In analyses using pedigree-based models, computations can be efficiently conducted using sparse matrix techniques due to the sparse nature of the coefficient matrix of the MME [14]. However, with the increase of genomic information and the emergence of high-dimensional datasets, the coefficient matrix of the MME tends to become denser. Consequently, animal and plant breeders working with large genomic data need to address this challenge of increased computational complexity.

To improve the computational efficiency and capability of REML analyses, various methods have been introduced. Masuda et al. [15] employed the supernodal methods to optimize the MME setting-up and trace computation in the AI-REML for genomic models and demonstrated that significant performance improvement was achieved compared to the original AI-REML. Recently, Meyer [16] found that, with principal components parameterization of the MME, the computing time of (co)variance components estimation for single-step genomic best linear unbiased prediction (ssGBLUP) using AI-REML can be substantially reduced. Matilainen et al. [17, 18] presented and implemented Monte Carlo (MC) approaches within the EM-REML and Newton-type methods. These approaches allowed the approximation of PEV/PEC without explicitly making or inverting the MME coefficient matrix. Instead, the approximated PEV/PEC were computed by generating MC samples from distributions identical to those of the original data and the current VC estimates. They showed that the MC-based REML methods using the preconditioned conjugate gradient (PCG) solver and the iteration on data method, is an efficient approach for VC estimation in large and complex models.

In the standard AI-REML framework, the Hessian matrix is replaced by an average information (AI) matrix, which is computed as the mean of both the observed and expected information matrices. This approach is widely adopted due to its simplicity, as it eliminates the need for intricate trace calculations required in both the observed and the expected information matrices. Instead, the AI matrix can be computed by solving the MME for each VC, with the data replaced by a working vector derived from the current random effect solutions [13]. However, this step can be computationally intensive for large systems (e.g. multi-trait random regression models), as it requires the construction of a work matrix obtained by repeatedly solving the MME with different right-hand sides (RHS). Thompson [19] identified this issue and proposed an alternative approach, which only requires solving an augmented MME to obtain essential values for AI-REML.

In this study, we highlight the importance of the augmented AI-REML method, as previously proposed by Thompson [19]. In particular, we demonstrate the advantages of the augmented AI-REML over the standard AI-REML. Unlike the standard AI-REML, the augmented

AI-REML can bypass the need to construct the working matrix (hereinafter referred to as the **T** matrix) and provide a more flexible framework to achieve the same information matrix with reduced computational cost. Furthermore, we illustrate how the augmented AI-REML differs from the standard AI-REML in the way it updates the VC estimates during each iteration (prototype code provided). We investigate the conditions under which the augmented AI-REML can significantly reduce computational requirements compared to the standard AI-REML. In addition, we explore different solving strategies (both direct and iterative) within the AI-REML framework to enhance computational efficiency. Therefore, the aims of this study were: (1) to present a computationally efficient algorithm of augmented AI-REML, which streamlines the VC estimation by solving an augmented MME only once per AI-REML iteration; and (2) to apply this approach in VC estimation in a general framework for a multi-trait GBLUP model.

Methods

Statistical model

Consider a multi-trait GBLUP model [20] for l traits:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Zu} + \mathbf{e} \tag{1}$$

where $\mathbf{y}$ is the vector of observations for the l traits, with n records for each trait, $\mathbf{b}$ is the vector of fixed effects, $\mathbf{u}$ is the vector of random genomic breeding values, and $\mathbf{e}$ is the vector of random residuals. The design matrices $\mathbf{X}$ and $\mathbf{Z}$ relate the observations to the fixed and random effects, respectively. The random effects $\mathbf{u}$ and $\mathbf{e}$ are assumed to be independent of each other: $\mathbf{u} \sim N(\mathbf{0}, \mathbf{G})$ and $\mathbf{e} \sim N(\mathbf{0}, \mathbf{R})$, where $\mathbf{G} = \mathbf{G}_0 \otimes \mathbf{G}_{rm}$, $\mathbf{G}_0$ is an $l \times l$ genetic variance covariance matrix, $\mathbf{G}_{\text{rm}}$ is a $q \times q$ marker-based genomic relationship matrix [20] with q equal to the number of genotyped individuals, $\mathbf{R} = \mathbf{R}_0 \otimes \mathbf{I}$, $\mathbf{R}_0$ is a $l \times l$ residual variance covariance matrix, assuming all traits are recorded for each individual, $\mathbf{I}$ is an identity matrix size of n, and $\otimes$ denotes the Kronecker product. We have assumed that all traits are observed for an individual. Furthermore, we have also assumed that all records to have the same residual covariance structure, i.e., a homogenous variance over individuals. These assumptions can easily be relaxed, but would unnecessarily complicate the following derivations.

When the (co)variance matrices $\mathbf{G}_0$ and $\mathbf{R}_0$ are known, the fixed and random effects can be solved using the MME as follows:

$$\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z} + \mathbf{G}^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{b}} \\ \widehat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{y} \end{bmatrix} \tag{2}$$

Let $\mathbf{C}$ be the coefficient matrix on the left-hand side (LHS) of the MME (2), $\mathbf{s}$ be the vector of fixed and random effects, i.e., $\mathbf{s}' = \begin{bmatrix} \mathbf{b}' & \mathbf{u}' \end{bmatrix}$, and $\mathbf{W} = \begin{bmatrix} \mathbf{X} & \mathbf{Z} \end{bmatrix}$. Let n_s be the total number of effects in $\mathbf{s}$. Then, the MME (2) can be written as $\mathbf{C}\widehat{\mathbf{s}} = \mathbf{W}'\mathbf{R}^{-1}\mathbf{y}$. Denote the vector of unknown VC by the parameter vector $\boldsymbol{\theta}' = \begin{bmatrix} \boldsymbol{\theta}'_{\text{G}} & \boldsymbol{\theta}'_{\text{R}} \end{bmatrix}$ where $\boldsymbol{\theta}_{\text{G}} = vech(\mathbf{G}_0)$, $\boldsymbol{\theta}_{\text{R}} = vech(\mathbf{R}_0)$, and $vech(\times)$ represents the operator extracting the unique elements from a symmetric matrix and reshape them into a vector form. In our case, the $\boldsymbol{\theta}$ vector of VC has v elements, containing $l(l+1)/2$ unique elements from $\mathbf{G}_0$ and $\mathbf{R}_0$. Henceforth, we denote the i-th genetic (co)variance in $\boldsymbol{\theta}_{\text{G}}$ as $\boldsymbol{\theta}_{\text{G}_i}$ and the i-th residual (co)variance in $\boldsymbol{\theta}_{\text{R}}$ as $\boldsymbol{\theta}_{\text{R}_i}$, respectively.

The standard AI REML

The REML log-likelihood function [21] can be written as follows:

$$\log\mathcal{L}(\boldsymbol{\theta}|\mathbf{y}) = const - \frac{1}{2}\log|\mathbf{V}| - \frac{1}{2}\log\left|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}\right| - \frac{1}{2}\mathbf{y}'\mathbf{P}\mathbf{y} \tag{3}$$

where $\mathbf{P} = \mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{W}\mathbf{C}^{-1}\mathbf{W}'\mathbf{R}^{-1}$, $\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \mathbf{R}$, and the constant *const* is independent of VC in $\boldsymbol{\theta}$. The REML estimates of $\widehat{\boldsymbol{\theta}}$ maximize the REML likelihood function $\log\mathcal{L}(\boldsymbol{\theta}|\mathbf{y})$ given the observed data.

Because complex models do not allow a closed-form solution of $\boldsymbol{\theta}$ that maximizes $\log\mathcal{L}(\boldsymbol{\theta}|\mathbf{y})$, iterative methods need to be used. The AI-REML [11, 12] updates VC estimates from iteration k-1 to iteration k using the formula:

$$\begin{aligned} \widehat{\boldsymbol{\theta}}^{[k]} &= \widehat{\boldsymbol{\theta}}^{[k-1]} + \boldsymbol{\Delta} \\ &= \widehat{\boldsymbol{\theta}}^{[k-1]} - \left[\mathbf{I}_{\text{A}}\left(\widehat{\boldsymbol{\theta}}^{[k-1]}\right)\right]^{-1}\mathbf{J}\left(\widehat{\boldsymbol{\theta}}^{[k-1]}\right) \end{aligned} \tag{4}$$

where $\widehat{\boldsymbol{\theta}}^{[k-1]}$ is the vector of current VC estimates, $\boldsymbol{\Delta}$ is the updating vector of VC estimates, $\mathbf{I}_{\text{A}}\left(\widehat{\boldsymbol{\theta}}^{[k-1]}\right)$ is the AI matrix at $\widehat{\boldsymbol{\theta}}^{[k-1]}$, and $\mathbf{J}\left(\widehat{\boldsymbol{\theta}}^{[k-1]}\right)$ is the vector of first derivatives of the REML log-likelihood (aka the gradient vector) with respect to $\boldsymbol{\theta}$ evaluated at $\widehat{\boldsymbol{\theta}}^{[k-1]}$. The AI matrix is

$$\mathbf{I}_{\text{A}}(\boldsymbol{\theta}) = \frac{1}{2}(\mathbf{I}_{\text{O}}(\boldsymbol{\theta}) + \mathbf{I}_{\text{E}}(\boldsymbol{\theta})) \tag{5}$$

where $\mathbf{I}_{\text{O}}(\boldsymbol{\theta})$ is the observed information matrix and $\mathbf{I}_{\text{E}}(\boldsymbol{\theta})$ is the expectation of the observed information matrix. Element (i,j), $i,j = 1,\ldots,v$, of $\mathbf{I}_{\text{O}}(\boldsymbol{\theta})$ is

$$\frac{\partial^2\log\mathcal{L}(\boldsymbol{\theta}|\mathbf{y})}{\partial\boldsymbol{\theta}_i\partial\boldsymbol{\theta}_j} = \mathbf{y}'\mathbf{P}\frac{\partial\mathbf{V}}{\partial\boldsymbol{\theta}_i}\mathbf{P}\frac{\partial\mathbf{V}}{\partial\boldsymbol{\theta}_j}\mathbf{P}\mathbf{y} - \frac{1}{2}tr(\mathbf{P}\frac{\partial\mathbf{V}}{\partial\boldsymbol{\theta}_i}\mathbf{P}\frac{\partial\mathbf{V}}{\partial\boldsymbol{\theta}_j}) \tag{6}$$

where $tr(\times)$ represents the matrix trace operator. Element (i,j), $i,j = 1,\ldots,v$, of $\mathbf{I}_{\text{E}}(\boldsymbol{\theta})$ is

$$E\left[\frac{\partial^2 \log \mathcal{L}(\boldsymbol{\theta}|\mathbf{y})}{\partial \boldsymbol{\theta}_i \partial \boldsymbol{\theta}_j}\right] = \frac{1}{2} tr\left(\mathbf{P}\frac{\partial \mathbf{V}}{\partial \boldsymbol{\theta}_i}\mathbf{P}\frac{\partial \mathbf{V}}{\partial \boldsymbol{\theta}_j}\right) \quad (7)$$

Hence, element (*i*,*j*) of the AI matrix is $\frac{1}{2}\mathbf{y}'\mathbf{P}\frac{\partial \mathbf{V}}{\partial \boldsymbol{\theta}_i}\mathbf{P}\frac{\partial \mathbf{V}}{\partial \boldsymbol{\theta}_j}\mathbf{P}\mathbf{y}$ [11, 13] and the AI matrix is

$$\begin{aligned}\mathbf{I}_{\mathrm{A}}(\boldsymbol{\theta}) =& \frac{1}{2}\mathbf{F}'\mathbf{P}\mathbf{F}\\ =& \frac{1}{2}\left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F} - \mathbf{F}'\mathbf{R}^{-1}\mathbf{W}\mathbf{C}^{-1}\mathbf{W}'\mathbf{R}^{-1}\mathbf{F}\right)\\ =& \frac{1}{2}(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F} - (\mathbf{C}^{-1}\mathbf{W}'\mathbf{R}^{-1}\mathbf{F})'\mathbf{W}'\mathbf{R}^{-1}\mathbf{F})\\ =& \frac{1}{2}(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F} - \mathbf{T}'\mathbf{W}'\mathbf{R}^{-1}\mathbf{F})\end{aligned} \quad (8)$$

where the working matrices $\mathbf{F} = \left[\begin{array}{ccc}\mathbf{f}_1 & \ldots & \mathbf{f}_v\end{array}\right]$ and $\mathbf{T} = \left[\begin{array}{ccc}\mathbf{t}_1 & \ldots & \mathbf{t}_v\end{array}\right]$ have v columns but n and n_s rows, respectively. Column i in $\mathbf{F}$ is [18]

$$\begin{aligned}\mathbf{f}_i =& \frac{\partial \mathbf{V}}{\partial \boldsymbol{\theta}_i}\mathbf{P}\mathbf{y}\\ =& \mathbf{Z}\frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}_i}\mathbf{G}^{-1}\widehat{\mathbf{u}} + \frac{\partial \mathbf{R}}{\partial \boldsymbol{\theta}_i}\mathbf{R}^{-1}\widehat{\mathbf{e}}\end{aligned} \quad (9)$$

where $\widehat{\mathbf{e}} = \mathbf{y} - \mathbf{X}\widehat{\mathbf{b}} - \mathbf{Z}\widehat{\mathbf{u}}$. Column i in $\mathbf{T}$ is the solution to the MME (2) but using $\mathbf{f}_i$ in place of the original $\mathbf{y}$ in the RHS (see Fig. 1 for illustration) [13]:

$$\mathbf{T} = \mathbf{C}^{-1}\mathbf{W}'\mathbf{R}^{-1}\mathbf{F} \quad (10)$$

The gradient vector $\mathbf{J}(\boldsymbol{\theta})$ element i, i=1,...,v, is

$$\frac{\partial \log \mathcal{L}(\boldsymbol{\theta}|\mathbf{y})}{\partial \boldsymbol{\theta}_i} = \frac{1}{2} tr\left(\mathbf{P}\frac{\partial \mathbf{V}}{\partial \boldsymbol{\theta}_i}\right) - \frac{1}{2}\mathbf{y}'\mathbf{P}\frac{\partial \mathbf{V}}{\partial \boldsymbol{\theta}_i}\mathbf{P}\mathbf{y} \quad (11)$$

Specifically, the gradient vector for the genetic VC is

$$\mathbf{J}(\boldsymbol{\theta}_{\mathrm{G}}) = -\frac{1}{2} vech(q\mathbf{G}_0^{-1} - \mathbf{G}_0^{-1}(\mathbf{L}_{\mathrm{G}} + \mathbf{U}'\mathbf{G}_{\mathrm{rm}}^{-1}\mathbf{U})\mathbf{G}_0^{-1}) \quad (12)$$

where q is the number of levels within trait in the random effects $\mathbf{u}$, i.e., the genomic breeding values in this study, $\mathbf{U}$ is a reshaped matrix of q by l of the $\mathbf{u}$ vector, and $\mathbf{L}_{\mathrm{G}}$ is an l by l matrix. The element (*i*,*j*) in $\mathbf{L}_{\mathrm{G}}$ is $tr\left(\mathbf{K}_{\mathrm{G},ij}\right)$ where $\mathbf{K}_{\mathrm{G},ij}$ is an q by q submatrix of $(\mathbf{I}_l \otimes \mathbf{G}_{\mathrm{rm}}^{-1})\mathbf{C}^{\mathbf{uu}}$ corresponding to the random effects of trait i and j, $\mathbf{C}^{\mathbf{uu}}$ is the submatrix of $\mathbf{C}^{-1}$ corresponding to the random effects $\mathbf{u}$ in MME (2); $\mathbf{I}_l$ is an l by l identity matrix, and $\otimes$ denotes the Kronecker product.

Correspondingly, the gradient vector for the residual VC is

$$J(\boldsymbol{\theta}_{\mathrm{R}}) = -\frac{1}{2} vech(n\mathbf{R}_0^{-1} - \mathbf{R}_0^{-1}(\mathbf{L}_{\mathrm{R}} + \mathbf{E}'\mathbf{E})\mathbf{R}_0^{-1}) \quad (13)$$

where n is the number of records, $\mathbf{E}$ is a reshaped matrix of residuals $\mathbf{e}$ of n by l, and $\mathbf{L}_{\mathrm{R}}$ is an l by l matrix. The element (*i*,*j*) in $\mathbf{L}_{\mathrm{R}}$ is $tr\left(\mathbf{K}_{\mathrm{R},ij}\right)$, where $\mathbf{K}_{\mathrm{R},ij}$ is an n by n submatrix of $\mathbf{W}\mathbf{C}^{-1}\mathbf{W}'$ corresponding to the traits i and j.

The augmented AI REML

The update vector $\boldsymbol{\Delta}$ of VC estimates in Eq. (4) is

$$\boldsymbol{\Delta} = -\left[\mathbf{I}_{\mathrm{A}}\left(\widehat{\boldsymbol{\theta}}^{[k-1]}\right)\right]^{-1}\mathbf{J}\left(\widehat{\boldsymbol{\theta}}^{[k-1]}\right) \quad (14)$$

This indicates that $\boldsymbol{\Delta}$ can be obtained by solving the following linear equations:

$$\left[\mathbf{I}_{\mathrm{A}}\left(\widehat{\boldsymbol{\theta}}^{[k-1]}\right)\right]\boldsymbol{\Delta} = -\mathbf{J}\left(\widehat{\boldsymbol{\theta}}^{[k-1]}\right) \quad (15)$$

Based on Eq. (8), Eq. (15) can be written as

$$\mathbf{F}'\mathbf{P}\mathbf{F}\boldsymbol{\Delta} = -2\mathbf{J}\left(\widehat{\boldsymbol{\theta}}^{[k-1]}\right) \quad (16)$$

In Eq. (16), the RHS for the genetic VC (G_i), which can be derived from Eq. (12), is

$$-2\mathbf{J}(\boldsymbol{\theta})_{G_i} = \widehat{\mathbf{u}}'\mathbf{G}^{-1}\left(\frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}_{G_i}}\right)\mathbf{G}^{-1}\widehat{\mathbf{u}} - \mathrm{t}_{G_i} \quad (17)$$

where $\mathrm{t}_{G_i} = tr[\mathbf{G}^{-1}(\frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}_{G_i}})\mathbf{G}^{-1}(\mathbf{G} - \mathbf{C}^{\mathbf{uu}})]$.

Because $\mathbf{P}\mathbf{y} = \mathbf{R}^{-1}\widehat{\mathbf{e}} = \mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{W}\widehat{\mathbf{s}}\right)$, with $\widehat{\mathbf{e}} = \mathbf{y} - \mathbf{W}\widehat{\mathbf{s}}$, and $\mathbf{G}^{-1}\widehat{\mathbf{u}} = \mathbf{Z}'\mathbf{P}\mathbf{y}$ [11], then

$$\mathbf{G}^{-1}\widehat{\mathbf{u}} = \mathbf{Z}'\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{W}\widehat{\mathbf{s}}\right) \quad (18)$$

Hence, Eq. (17) can be written as

$$-2\mathbf{J}(\boldsymbol{\theta})_{G_i} = \mathbf{f}_i'\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{W}\widehat{\mathbf{s}}\right) - \mathbf{t}_{G_i} \quad (19)$$

because of Eq. (18) and $\mathbf{f}_i = \mathbf{Z}(\frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}_{G_i}})\mathbf{G}^{-1}\widehat{\mathbf{u}}$ according to Eq. (9). Similarly, the RHS for the residual VC (R_i), which can be derived from Eq. (13), is

$$-2\mathbf{J}(\boldsymbol{\theta})_{R_i} = \widehat{\mathbf{e}}'\mathbf{R}^{-1}\left(\frac{\partial \mathbf{R}}{\partial \boldsymbol{\theta}_{R_i}}\right)\mathbf{R}^{-1}\widehat{\mathbf{e}} - \mathrm{t}_{R_i} \quad (20)$$

where $\mathrm{t}_{R_i} = tr[(\frac{\partial \mathbf{R}}{\partial \boldsymbol{\theta}_{R_i}})\mathbf{R}^{-1}] - \mathrm{tr}[(\frac{\partial \mathbf{C}}{\partial \boldsymbol{\theta}_{R_i}})\mathbf{C}^{-1}]$. Thus, Eq. (20) can be written as

$$-2\mathbf{J}(\boldsymbol{\theta})_{R_i} = \mathbf{f}_i'\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{W}\widehat{\mathbf{s}}\right) - \mathbf{t}_{R_i} \quad (21)$$

Combining Eq. (19) and Eq. (21) gives the RHS in Eq. (16) as

$$-2\mathbf{J}\left(\widehat{\boldsymbol{\theta}}^{[k-1]}\right) = \mathbf{F}'\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{W}\widehat{\mathbf{s}}\right) - \mathbf{t} \quad (22)$$

where $\mathbf{t}' = \left[\begin{array}{cc}\mathbf{t}_G' & \mathbf{t}_R'\end{array}\right]$. Thus, using Eq. (8), Eq. (15) can be expressed as

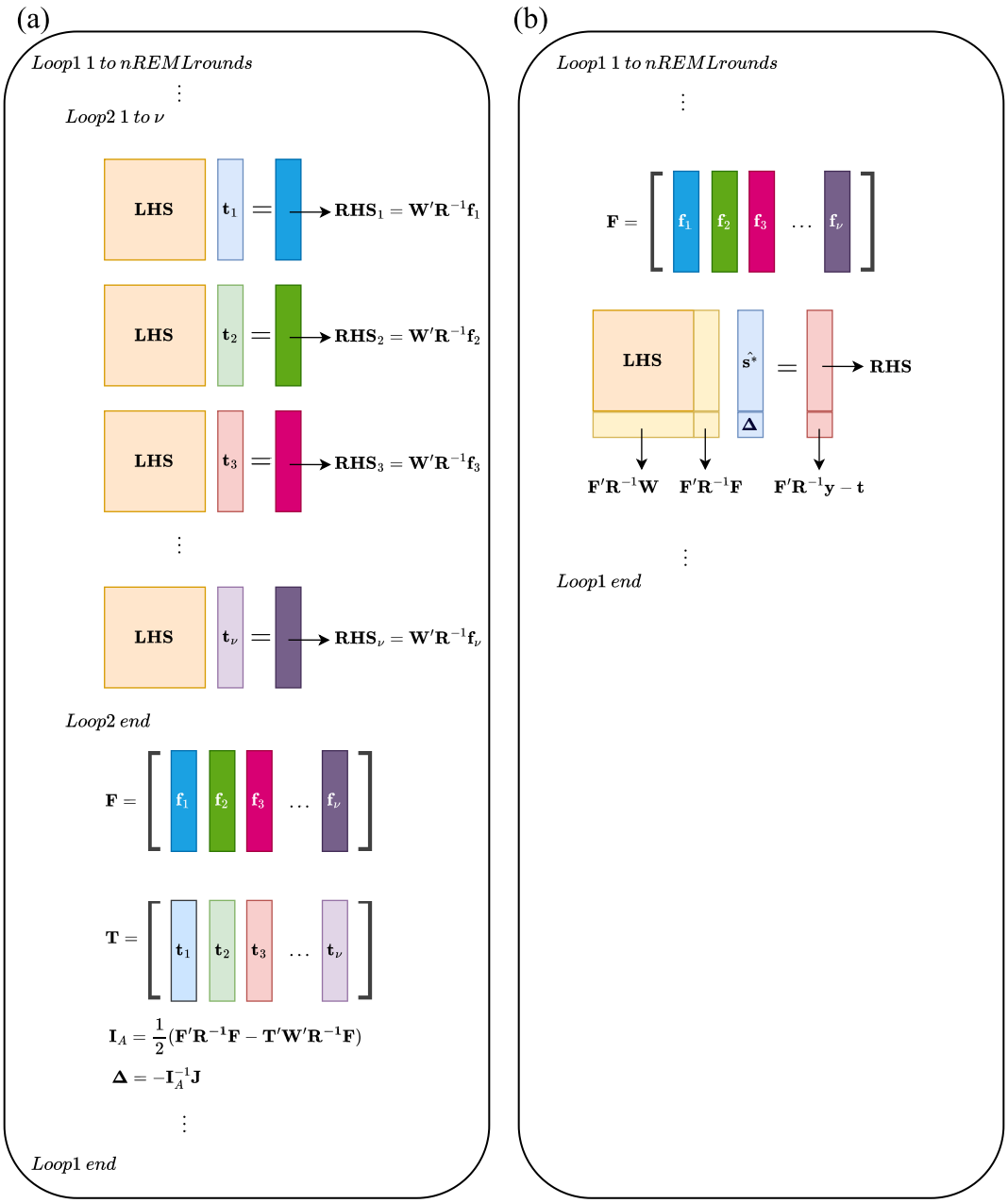

Fig. 1 Illustration of the standard AI-REML. **a** vs. the augmented AI-REML **b** for (co)variance component (VC) estimation. The standard AI-REML requires solving the mixed model equations (MME) for each VC to obtain the working vector ($\mathbf{t}_{i=1,\ldots,\nu}$), with the right-hand side (RHS) replaced by a suitable working vector ($\mathbf{f}_{i=1,\ldots,\nu}$), where n is the total number of VC in the model, $\mathbf{W} = \begin{bmatrix} \mathbf{X} & \mathbf{Z} \end{bmatrix}$, $\mathbf{R} = \mathbf{R}_0 \otimes \mathbf{I}$, $\mathbf{R}_0$ is a $l \times l$ residual variance covariance matrix, assuming all traits (l) are recorded for each individual, $\mathbf{I}$ is an identity matrix size of n/l, with n equal to the number of observations, and $\otimes$ denotes the Kronecker product. Note that the left-hand side (LHS) of the MME is unchanged for each solving process. $\mathbf{F}$ and $\mathbf{T}$ are working matrices containing column vectors of $\mathbf{f}_{i=1,\ldots,\nu}$ and $\mathbf{t}_{i=1,\ldots,\nu}$, respectively. The average information matrix $\mathbf{I}_A$ is computed using $\mathbf{F}$ and $\mathbf{T}$, then the updating vector based on the current VC estimates ($\boldsymbol{\Delta}$) is computed using the inverse of $\mathbf{I}_A$ and gradient vector ($\mathbf{J}$). For the augmented AI-REML, the updating vector $\boldsymbol{\Delta}$ can be solved using an MME where the original MME and model has been augmented by an effect $\boldsymbol{\Delta}$ with the working matrix $\mathbf{F}$ and RHS for this effect has been corrected by the trace terms ($\mathbf{t}$) in the first derivatives of the REML log-likelihood

$$\left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F} - \mathbf{F}'\mathbf{R}^{-1}\mathbf{W}\mathbf{C}^{-1}\mathbf{W}'\mathbf{R}^{-1}\mathbf{F}\right)\boldsymbol{\Delta} = \mathbf{F}'\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{W}\hat{\mathbf{s}}\right) - \mathbf{t} \quad (23)$$

Because $\mathbf{C}\hat{\mathbf{s}} = \mathbf{W}'\mathbf{R}^{-1}\mathbf{y}$, the part involving $\hat{\mathbf{s}}$ in the RHS of Eq. (23) can be reformulated:

$$\mathbf{F}'\mathbf{R}^{-1}\mathbf{W}\hat{\mathbf{s}} = \mathbf{F}'\mathbf{R}^{-1}\mathbf{W}\mathbf{C}^{-1}\mathbf{W}'\mathbf{R}^{-1}\mathbf{y} \quad (24)$$

This allows expressing Eq. (23) in an augmented form [19] as follows:

$$\begin{bmatrix} \mathbf{C} & \mathbf{W}'\mathbf{R}^{-1}\mathbf{F} \\ \mathbf{F}'\mathbf{R}^{-1}\mathbf{W} & \mathbf{F}'\mathbf{R}^{-1}\mathbf{F} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{s}}^* \\ \boldsymbol{\Delta} \end{bmatrix} = \begin{bmatrix} \mathbf{W}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{F}'\mathbf{R}^{-1}\mathbf{y} - \mathbf{t} \end{bmatrix} \quad (25)$$

Thus, the updating vector $\boldsymbol{\Delta}$ can be solved using an augmented MME, where $\hat{\mathbf{s}}^*$ is the new solution vector of fixed and random effects. The original MME and model are augmented by "the update effect $\boldsymbol{\Delta}$" with the working matrix $\mathbf{F}$ as its model matrix. RHS of the update effects, i.e., $\mathbf{F}'\mathbf{R}^{-1}\mathbf{y}$, is corrected by the trace terms $\mathbf{t}$ in the first derivatives of the REML log-likelihood (see Fig. 1 for illustration).

Equation (23) allows solving the updating vector $\boldsymbol{\Delta}$ without the need to make the augmented MME (25). Alternatively, it is possible to avoid solving the entire MME (25) by absorbing the augmented part into the original MME when employing a direct solver during each REML iteration. Thus, to obtain the solutions for $\boldsymbol{\Delta}$, the computational cost can be kept low by solving:

$$\mathbf{LHS}^*\boldsymbol{\Delta} = \mathbf{RHS}^* \quad (26)$$

where

$$\mathbf{LHS}^* = \mathbf{F}'\mathbf{R}^{-1}\mathbf{F} - \mathbf{F}'\mathbf{R}^{-1}\mathbf{W}\mathbf{C}^{-1}\mathbf{W}'\mathbf{R}^{-1}\mathbf{F} \quad (27)$$

$$\mathbf{RHS}^* = \mathbf{F}'\mathbf{R}^{-1}\mathbf{y} - \mathbf{t} - \mathbf{F}'\mathbf{R}^{-1}\mathbf{W}\mathbf{C}^{-1}\mathbf{W}'\mathbf{R}^{-1}\mathbf{y} \quad (28)$$

and in both Eq. (27) and Eq. (28), the term $\mathbf{W}\mathbf{C}^{-1}\mathbf{W}'$ is precomputed. Note that the dimension of **LHS*** (26) is ν by ν, corresponding to the augmented part only.

Data simulation

Data were simulated over 10 generations after the base population using the AlphaSimR package [22]. The simulation had five traits. The cattle species history was used for generating the base population haplotypes with an effective population size of 200. The genome consisted of 30 chromosomes. The simulated traits were determined by 900 QTL, i.e., 30 QTL per chromosome. The QTL effects for all traits were simulated from the Gamma density with shape 0.4 and scale 1.0.

After the historical population simulation, a base population of 1000 males and 1000 females was generated. The base population individuals were mated randomly, each mating producing one offspring. After the base population, the breeding population was created by selecting the top 100 males and 1000 females from the base population and the newly generated offspring to form the breeding population. The selection was based on a phenotypic index of all traits weighing them equally. Each mating in the breeding population produced one offspring: either male or female at equal numbers. In every subsequent generation, the best 100 males and 1000 females were selected from the group consisting of the current breeding population and the offspring produced by the random mating of the breeding animals.

The final pedigree consisted of 6100 females and 6100 males after simulating the breeding programme for 10 generations. In the simulation, every individual was simulated to have one observation from all correlated traits. All individuals in the pedigree were genotyped with a total number of 54,000 single nucleotide polymorphisms (SNPs). The VC used to simulate these five traits were from Nordic Cattle Genetic Evaluation (NAV) used for the evaluation of metabolic body weight (metabolic body weight during the first, second, and third lactation, stature, and carcass weight) [23]. Table 1 presents the genetic (co)variances, heritability, and genetic correlations between these five traits.

Analyses

The multi-trait GBLUP model (1) with the general means, genetic effects, and residuals was used to fit the simulated

Table 1 Genetic (co)variances (lower triangular elements), heritability (h^2), and genetic correlations (elements above diagonal) used in the simulation for five traits

Trait	1	2	3	4	5	h^2
1	*27.60*	0.97	0.95	0.65	0.77	0.46
2	31.00	*37.00*	0.98	0.70	0.84	0.50
3	34.64	41.37	*48.16*	0.68	0.85	0.56
4	10.30	12.80	14.16	*9.00*	0.59	0.60
5	117.90	148.76	171.80	61.60	*847.60*	0.52

data. The genomic relationship matrix was constructed using VanRaden's method 1 [20] with the allele frequencies computed from the genotyped data. We investigated VC estimation as a function of the number of VC in the model, including two-trait, three-trait, four-trait, and five-trait GBLUP models, representing 6, 12, 20, and 30 VC, respectively.

Both the augmented AI-REML and the standard AI-REML methods were applied to all models. Within each REML iteration, we used the direct solving method via inverting the coefficient matrix of the MME. In addition, for the two-, three-, and four-trait GBLUP models, we utilized the PCG solver as an iterative solving method to assess the augmented AI-REML; however, for simplicity, the traces were still obtained by inversion. The convergence criteria in the PCG solver was $\|\mathbf{Cs}-\mathbf{r}\| < 10^{-5}$ where $\mathbf{C}$ is the coefficient matrix of the MME, $\mathbf{r}$ is the right-hand-side vector, $\mathbf{s}$ is the solution vector, and $\|.\|$ is the Euclidean norm of a vector.

For all analyses, identity matrices were used as the initial values for the VC. The convergence indicator in both AI-REML methods was based on the relative change between the current and previous iteration of VC estimates, i.e., $\frac{\left(\widehat{\boldsymbol{\theta}}^{[k]}-\widehat{\boldsymbol{\theta}}^{[k-1]}\right)'\left(\widehat{\boldsymbol{\theta}}^{[k]}-\widehat{\boldsymbol{\theta}}^{[k-1]}\right)}{\widehat{\boldsymbol{\theta}}^{[k]'}\widehat{\boldsymbol{\theta}}^{[k]}}$. The threshold value was set to 1.0E-12. To compare computational efficiency, we evaluated the augmented AI-REML against the standard AI-REML in terms of the elapsed computing time per iteration. All analyses were carried out using the Julia programming language [24] on a Linux server with an Intel(R) Xeon(R) Gold 6248 CPU (2.5 GHz) and 1.5 TB RAM.

Results

Overall, both the augmented AI-REML and the standard AI-REML produced identical VC estimates and used the same number of iterations until convergence. Table 2 shows the elapsed time per REML iteration using the direct solving method for the augmented AI-REML and the standard AI-REML across two-, three-, four-, and five-trait GBLUP models. Our analyses of the simulated datasets revealed tangible improvements in computational efficiency with the augmented AI-REML, leading to reductions in computing time per REML iteration. Although the augmented AI-REML and the standard AI-REML required similar computing times for models with a small number of VC (such as the two- and three-trait GBLUP model), the augmented AI-REML demonstrated more notable reductions in computing time as the number of VC in the model increased. The largest reduction in computing time was observed in the five-trait GBLUP model with 30 VC. Peak core memory usage was comparable between the augmented and the standard AI-REML methods.

Table 3 presents the elapsed time per REML iteration using the iterative solving method for both the augmented AI-REML and the standard AI-REML across two-, three-, and four-trait GBLUP models. In contrast to the direct solving method, both the augmented AI-REML and the standard AI-REML exhibited longer computing times when using the iterative solving method, especially for models with many VC, such as the four-trait GBLUP model. However, the augmented AI-REML demonstrated substantial improvements in computational efficiency compared to the standard AI-REML by eliminating the need to solve the MME for each VC. Based on the analysis of the simulated datasets, the elapsed time of each REML iteration was reduced by 75%, 84%, and 86% for the two-, three-, and four-trait GBLUP models, respectively.

Table 2 Elapsed times (seconds) for (co)variance component (VC) estimations using augmented and standard average information restricted maximum likelihood (AI-REML) with direct solver for two-, three-, four-, and five-trait genomic best linear unbiased prediction (GBLUP)

Model	Two-trait	Three-trait	Four-trait	Five-trait
v[a]	6	12	20	30
N_{eq}[b]	24,402	36,603	48,804	61,005
N_{it}[c]	14	15	15	17
Standard AI-REML (s)	216	677	1554	2972
Augmented AI-REML (s)	215	674	1523	2885

[a] Number of VC in the mixed model, [b]Number of equations in the mixed model, [c]Number of REML iterations to achieve convergence

Table 3 Elapsed times (seconds) for (co)variance component (VC) estimations using augmented and standard average information restricted maximum likelihood (AI-REML) with iterative solver for two-, three-, and four-trait genomic best linear unbiased prediction (GBLUP)

Model	Two-trait	Three-trait	Four-trait
v[a]	6	12	20
N_{eq}[b]	24,402	36,603	48,804
N_{it}[c]	14	15	15
Standard AI-REML (s)	2091	9780	33,452
Augmented AI-REML (s)	528	1569	3266

[a] Number of VC in the mixed model, [b]Number of equations in the mixed model, [c]Number of REML iterations to achieve convergence

Discussion

In this paper, we have introduced a computationally efficient AI-REML algorithm called augmented AI-REML. While the standard AI-REML is known for its rapid convergence, it has some computational challenges, particularly when utilizing the iterative solver during AI-REML iterations. Specifically, the standard AI-REML requires solving the MME for each VC, which becomes increasingly resource-intensive as the number of VC grows. In contrast, the augmented AI-REML algorithm streamlines the computational process by solving an augmented MME only once. This novel approach offers both computational simplicity and efficiency. Notably, this study represents the first implementation of the augmented AI-REML method. Our results highlight its superiority, especially when estimating a large number of VC in the model.

A typical AI-REML algorithm relies on elements from the inverse coefficient matrix of the MME to compute the trace terms. Consequently, it has become common practice for AI-REML applications to use the direct solving method to compute the inverse. However, as the use of genomic information increases, the coefficient matrix of the MME becomes denser, posing computational challenges when analyzing large genomic datasets. Therefore, enabling VC estimation by the AI-REML method for large datasets and accelerating its computational process remains a critical concern in the genomic era. Masuda et al. [15] developed a package called YAMS. YAMS enhances the MME setup, reorders sparse structures for trace computations, and enables parallel computing for large dense blocks. They reported that the performance of YAMS was on average 10 times faster than FSPAK, a sparse matrix operation package based on traditional pedigree-based models. Laporte et al. [25] introduced the Min–Max (MM) algorithm as an alternative to the classical AI-REML algorithm for VC estimation in plant breeding. Although their method requires deriving a surrogate function within each iteration, it can offer a promising computational speed-up. Meyer [16] proposed a computational strategy involving the reparameterization of the MME to principal components. Her approach takes into account differences in the sparsity of the coefficient matrix within the single-step GBLUP model. She demonstrated a substantial reduction in computing time per iteration by leveraging this transformation to principal components.

In the current study, we first applied the direct solving method to both the augmented and the standard AI-REML. Given that the inverse of the coefficient matrix was precomputed and stored in memory, solving the MME multiple times within each iteration via multiplication with the corresponding vector on the right-hand side did not result in extreme computational expense. Consequently, based on the current datasets, the reduction in computing time per iteration using augmented AI-REML was not significantly different from the standard AI-REML (Table 2). Moreover, parallelization techniques such as OpenMP can be employed to parallelize the MME solving step when dealing with a large number of VC.

Consider a genomic model such as GBLUP used in the current study, which produces a dense coefficient matrix for the MME. When using the direct solver, the computational cost of the augmented AI-REML can be further reduced by solving a small linear system (Eq. (26)) with a size equal to the number of VC to be estimated in the model (ν). From a theoretical perspective, the estimate of computational cost in terms of the floating point operations per second (FLOPS) can be reduced from $\frac{1}{3}(n+\nu)^3 + 2(n+\nu)^2$ to $\frac{1}{3}\nu^3 + 2\nu^2$. Note that the FLOPS for the standard AI-REML in Eq. (10) is $\nu(2n^2 - n)$, where n is the number of equations in the linear mixed model. This advantage of the augmented AI-REML is due to the precomputation of $\mathbf{WC}^{-1}\mathbf{W}'$, which can be efficiently reused when absorbing the augmented portion of the MME into the original one. However, it is crucial to recognize that this feature cannot be preserved when using an iterative solver, because no inverted coefficient matrix of the MME is available. Consequently, the entire augmented linear system (Eq. (25)) must be solved. Another noteworthy aspect of the augmented AI-REML is that, during each iteration, it avoids computing quadratic terms, such as $\mathbf{u}'\mathbf{G}_{\mathrm{rm}}^{-1}\mathbf{u}$ and $\mathbf{e}'\mathbf{e}$ in our examples, even though the computation time for these terms is negligible.

Iterative solving methods such as PCG, in combination with iteration on data techniques [26–28], are the preferred approach for solving large MME when computing or storing the Cholesky factor of the MME is infeasible. The advantage of the iteration on data approach lies in its avoidance of explicit construction of the MME. In a study by Matilainen et al. [18], they implemented an MC algorithm within the standard AI-REML framework called MC (standard) AI-REML. In their approach, the MC samples generated from the same distribution as the original model were used to approximate PEV/PEC as equivalent to the inverse of the coefficient matrix of the MME. The MC-based REML methods improve the capability to handle large-scale VC estimation. However, it is essential to note that the MC (standard) AI-REML requires solving the MME for each VC in the model during every iteration. Consequently, the MC AI-REML method imposes a significant computational burden, especially when dealing with complex models and large datasets, such as multi-trait random regression models.

The augmented AI-REML can offer a significant advantage over the standard AI-REML in terms of computational efficiency, particularly for large multi-trait genomic models. In the standard AI-REML approach, the inverse of the MME can be used to solve the update vector of AI-REML as in Eq. (10). In addition, the inverse can also provide the PEV/PEC values required in Eq. (12) and Eq. (13). However, when an MC approach is used, the update vector and the PEV/PEC values need to be computed separately. Consequently, the standard MC AI-REML method is computationally less attractive than the EM-REML method where only one MME solving for the update is needed [18]. The augmented AI-REML can give a significant reduction in computing time in MC AI-REML because the augmented MME need to be solved only once to obtain the update vector within each REML iteration. This increases the effectiveness of MC AI-REML and will make it an attractive approach because AI-REML often converges in fewer iterations than EM-REML. As shown in Table 3, there is a substantial reduction in the computing time per REML iteration with the augmented AI-REML. This indicates the benefit of combining the MC method with the augmented AI-REML for large-scale VC estimation.

In this study, we focused on demonstrating the augmented AI-REML algorithm in analyses of a simple multi-trait GBLUP model. This algorithm can offer computational feasibility and simplicity across various models and can be integrated into existing AI-REML applications. The reduction in computing time achieved with the augmented AI-REML depends on the dimension of the dataset and the chosen model. However, it is important to recognize that the augmented AI-REML algorithm does not improve the convergence rates. In other words, it provides identical estimates and converges within the same number of iterations as the standard AI-REML. Moreover, in our analyses, even when using the PCG solver in both the augmented and the standard AI-REML algorithms, the trace terms were still derived by brute force inversion of the coefficient matrix of the MME, rather than approximated by the MC method.

Conclusions

In this study, we introduced and demonstrated the augmented AI-REML algorithm, which is designed to improve the computational efficiency of VC estimation by AI-REML. We compared the augmented AI-REML with the standard AI-REML, employing both direct and iterative solvers in the AI-REML algorithms. In particular, the direct solver resulted in worthwhile reductions in computing time, while the iterative solver achieved significant time savings. The reductions were larger when more VC were estimated in the model. However, further research is needed to study the effect of a larger number of estimated variance components on the convergence of the iterative method for solving the augmented system. Our results underscore the potential utility of augmented AI-REML as an appealing approach for large-scale VC estimation in the genomic era.

Acknowledgements
Not applicable

Author contributions
HG, EM, and IS conceived the study design; HG wrote the code and conducted the analyses; ML provided the (co)variance components used for the simulation; RT contributed to the evaluation and discussion of the methods. HG wrote the first draft and all authors provided valuable insights throughout the writing process. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
R code used to generate the simulated data and Julia code for the augmented and standard AI. REML can be found at https://github.com/hongdinggao/AI-REML.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 10 June 2024 Accepted: 28 October 2024
Published online: 21 November 2024

References

1. Hidalgo J, Tsuruta S, Lourenco D, Masuda Y, Huang Y, Gray KA, et al. Changes in genetic parameters for fitness and growth traits in pigs under genomic selection. J Anim Sci. 2020;98:1–12.
2. Gao H, Madsen P, Aamand GP, Thomasen JR, Sorensen AC, Jensen J. Bias in estimates of variance components in populations undergoing genomic selection: a simulation study. BMC Genomics. 2019;20:956.
3. Misztal I, Bradford HL, Lourenco DAL, Tsuruta S, Masuda Y, Legarra A, et al. Studies on inflation of GEBV in single-step GBLUP for type. Interbull Bull. 2017;51:38–42.
4. Wiggans GR, VanRaden PM, Cooper TA. Technical note: adjustment of all cow evaluations for yield traits to be comparable with bull evaluations. J Dairy Sci. 2012;95:3444–7.
5. Misztal I, Lourenco D, Legarra A. Current status of genomic evaluation. J Anim Sci. 2020;98: skaa101.
6. Jensen J. Estimation of genetic variance in the age of genomics. J Anim Breed Genet. 2016;133:333.
7. Patterson HD, Thompson R. Recovery of inter-block information when block sizes are unequal. Biometrika. 1971;58:545–54.
8. Meyer K. Estimating variances and covariances for multivariate animal models by restricted maximum likelihood. Genet Sel Evol. 1991;23:67–83.
9. Smith AB, Cullis BR, Thompson R. The analysis of crop cultivar breeding and evaluation trials: an overview of current mixed model approaches. J Agric Sci. 2005;143:449–62.

10. Hofer A. Variance component estimation in animal breeding: a review. J Anim Breed Genet. 1998;115:247–65.
11. Johnson DL, Thompson R. Restricted maximum likelihood estimation of variance components for univariate animal models using sparse matrix techniques and average information. J Dairy Sci. 1995;78:449–56.
12. Gilmour AR, Thompson R, Cullis BR. Average information REML: an efficient algorithm for variance parameter estimation in linear mixed models. Biometrics. 1995;51:1440–50.
13. Jensen J, Mäntysaari EA, Madsen P, Thompson R. Residual maximum likelihood estimation of (co)variance components in multivariate mixed linear models using average information. J Indian Soc Agric Stat. 1997;49:215–36.
14. Misztal I, Perez-Enciso M. Sparse matrix inversion for restricted maximum likelihood estimation of variance components by expectation-maximization. J Dairy Sci. 1993;76:1479–83.
15. Masuda Y, Aguilar I, Tsuruta S, Misztal I. Technical note: acceleration of sparse operations for average-information REML analyses with supernodal methods and sparse-storage refinements. J Anim Sci. 2015;93:4670–4.
16. Meyer K. Reducing computational demands of restricted maximum likelihood estimation with genomic relationship matrices. Genet Sel Evol. 2023;55:7.
17. Matilainen K, Mantysaari EA, Lidauer MH, Stranden I, Thompson R. Employing a Monte Carlo algorithm in expectation maximization restricted maximum likelihood estimation of the linear mixed model. J Anim Breed Genet. 2012;129:457–68.
18. Matilainen K, Mäntysaari EA, Lidauer MH, Strandén I, Thompson R. Employing a Monte Carlo algorithm in Newton-type methods for restricted maximum likelihood estimation of genetic parameters. PLoS ONE. 2013;8: e80821.
19. Thompson R. Desert island papers—a life in variance parameter and quantitative genetic parameter estimation reviewed using 16 papers. J Anim Breed Genet. 2019;136:230–42.
20. VanRaden PM. Efficient methods to compute genomic predictions. J Dairy Sci. 2008;91:4414–23.
21. Harville DA. Bayesian inference for variance components using only error contrasts. Biometrika. 1974;61:383.
22. Chris Gaynor R, Gorjanc G, Hickey JM. AlphaSimR: an R package for breeding program simulations. G Genes Genomes Genetics. 2021;11: jkaa017.
23. Mehtiö T, Pitkänen T, Leino AM, Mäntysaari EA, Kempe R, Negussie E, et al. Genetic analyses of metabolic body weight, carcass weight and body conformation traits in Nordic dairy cattle. Animal. 2021;15: 100398.
24. Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: a fresh approach to numerical computing. SIAM Rev. 2017;59:65–98.
25. Laporte F, Charcosset A, Mary-Huard T. Efficient ReML inference in variance component mixed models using a Min-Max algorithm. PLoS Comput Biol. 2022;18: e1009659.
26. Misztal I, Gianola D. Indirect solution of mixed model equations. J Dairy Sci. 1987;70:716–23.
27. Strandén I, Lidauer M. Solving large mixed linear models using preconditioned conjugate gradient iteration. J Dairy Sci. 1999;82:2779–87.
28. Schaeffer LR, Kennedy BW. Computing strategies for solving mixed model equations. J Dairy Sci. 1986;69:575–9.

Publisher's Note