Acta Scientiarum

http://www.periodicos.uem.br/ojs/
ISSN on-line: 1807-8621
Doi: 10.4025/actasciagron.v46i1.64243

Genomic prediction using the lmekin function from the coxme R package

Clemeson Silva de Souza[1], Vinicius Silva dos Santos[2*] and Sebastião Martins Filho[3]

[1]Programa de Pós-graduação *Lato Sensu* em Estatística, Universidade Federal do Acre, Rodovia BR-364, km 04, 69920-900, Rio Branco, Acre, Brazil. [2]Centro de Ciências Exatas e Tecnológicas, Universidade Federal do Acre, Rio Branco, Acre, Brazil. [3]Departamento de Estatística, Universidade Federal de Viçosa, Viçosa, Minas Gerais, Brazil. *Author for correspondence. E-mail: vinicius.santos@ufac.br

ABSTRACT. The increasing use of genomic selection (GS) in plant and animal breeding programs has led to the development of software that fits models based on unique scenarios. Accordingly, several R packages have been developed for GS. The lmekin function from the coxme R package was one of the first functions implemented in R to fit models with random family effects using the pedigree–based relationship matrix. The function allows the user to provide the covariance structures for the random effects; thus, the GBLUP model can be fitted. This fitting process consists of replacing, in the traditional BLUP model, the additive relationship matrix derived from a pedigree by the additive relationship matrix derived from markers. Thus, the objective of this study was to employ the lmekin function in the context of genomic prediction by comparing the results of this function with those obtained using five R packages for GS: rrBLUP, BGLR, sommer, lme4qtl, and lme4GS. The comparisons were performed considering the computational times and predicted values for a wheat dataset and simulated big data. In addition, we implemented a 5-fold cross-validation scheme through considering the values predicted by the lmekin function for the wheat dataset. The results indicated that the lmekin function was effective in predicting genomic breeding values considering multiple random effects and relatively small sample sizes. The rrBLUP package processed the fastest for the scenario with only one genetic random effect, and the high temporal efficiency of the sommer package was confirmed for the scenario with more than one genetic random effect. Differences in computational times occurred because of the different algorithms implemented in the packages to estimate the variance components.

Keywords: mixed models; GBLUP; genomic relationship matrix; pedigree; genetic breeding.

Received on July 7, 2022.
Accepted on January 18, 2023.

Introduction

The superiority of the genomic selection (GS) method, proposed by Meuwissen, Hayes, and Goddard (2001), in relation to phenotypic selection has been confirmed in several plant and animal breeding programs (Resende, Silva, & Azevedo, 2014; Budhlakoti et al., 2022). This superiority has mainly resulted from a reduction in generation intervals, as young plants can be genotyped early in life and their genomic breeding values are computed for selection purposes, resulting in an improved selection accuracy. Moreover, GS considers the real genetic relationship of individuals instead of the average pedigree-based relationship (Mrode, 2014; Resende et al., 2014).

In genomic prediction models, the number of regressor variables (p), i.e., the number of markers, usually vastly exceeds the number of observations (n), which leads to the large "p" and small "n" problem (p >> n). Thus, GS requires methods that use some type of variable selection or shrinkage estimation procedures (de los Campos, Hickey, Pong-Wong, Daetwyler, & Calus, 2013; Resende et al., 2014; Budhlakoti et al., 2022). Alternatively, a mixed-model methodology can be employed within the context of GS. One of the most widely used methods consists of replacing, in the traditional BLUP model, the pedigree-based relationship matrix (A) with the marker-based relationship matrix (G), resulting in a method called the Genomic Best Linear Unbiased Prediction (GBLUP) (VanRaden, 2008; Yang et al., 2010; Mrode, 2014; Resende et al., 2014).

Genomic selection methods based on mixed models can be generalized to comparatively complex scenarios; for example, the prediction of additive and non-additive genetic effects using relationship information derived from markers and pedigree (Muñoz et al., 2014). Despite all computational advances,

there are few open-source statistical software that allow genomic predictions considering multiple random effects (Covarrubias-Pazaran, 2016).

According to Caamal-Pat et al. (2021), the first R packages developed for GS were BLR (de los Campos et al., 2009) and rrBLUP (Endelman, 2011). Motivated by the need to implement various genomic methods using Bayesian regression in a single program, Pérez and de los Campos (2014) developed the BGLR package, which supports models for continuous (censored or non-censored) and categorical (binary or ordinal multinomial) traits. To predict genomic breeding values incorporating more than one variance component in addition to the residual error, Covarrubias-Pazaran (2016) implemented the Sommer R-package, which is based on the Henderson mixed model equations. Other programs have implemented GS methods that utilize free software with an intuitive interface and without command lines (Cruz, 2016; Resende, 2016; Azevedo et al., 2019).

In R software, mixed model analyses have been performed mainly using the nlme (Pinheiro & Bates, 2000) and lme4 (Bates, Mächler, Bolker, & Walker, 2015) packages. However, neither package fits models with random genetic effects. One of the first functions implemented in R to fit such linear models is the lmekin (*linear mixed models with kinship*) function from the coxme package (Therneau, 2020), which is an extension of the lme function from the nlme R-package. This function has been widely used in human genetic studies but has few applications in plant and animal breeding.

As an extension of the lme4 R-package, to fit mixed models applied to animal breeding, Vazquez, Bates, Rosa, Gianola, and Weigel (2010) developed the pedigreemm R-package, which fits generalized linear models with pedigree-based information. However, the package does not allow the user to provide a variance–covariance matrix for genetic effects, making its use in GS unfeasible. Recently, two R packages have been developed, lme4qtl (Ziyatdinov et al., 2018) and lme4GS (Caamal-Pat et al., 2021), which both represent extensions of the lme4 package and fit GBLUP-like models with multiple random effects.

Unlike the pedigreemm R-package, the lmekin function allows the user to input a covariance matrix for random genetic effects, such as the genomic relationship matrix. Furthermore, the function allows for estimating multiple variance components, making it possible to fit the GBLUP model with multiple random effects.

The sommer package was compared with the rrBLUP and BGLR packages, and with four other R packages: ASReml-R, regress, EMMREML, and MCMCglmm (Covarrubias-Pazaran, 2016). The lme4qtl package was compared with the SOLAR package and the lmekin function, considering a genome-wide association study (GWAS) and not the prediction context in GS (Ziyatdinov et al., 2018). The lme4GS package was compared with the BGLR and sommer packages (Caamal-Pat et al., 2021). However, to the best of our knowledge, no study has compared the computational performances of these packages when fitting GBLUP-type models. Thus, the objective of this study was to employ the lmekin function in the context of genomic prediction by comparing the results of this function with those obtained using five different R packages for GS: rrBLUP, BGLR, sommer, lme4qtl, and lme4GS.

Material and methods

The lmekin function fits the general linear mixed model as follows (Therneau, 2020):

$$y = X\beta + Zb + e, \quad (1)$$

where y is the vector of observed values, β is the vector of fixed effects, $b \sim N(0, \sigma_e^2 D(\theta))$ is the vector of random effects, $e \sim N(0, I\sigma_e^2)$ is the random error vector, and X and Z are incidence matrices for fixed and random effects, respectively. The relative variance matrix D for random effects depends on the vector of the unknown variance parameters $\theta = (\sigma_1^2, \ldots, \sigma_1^2)/\sigma_e^2$. The model also assumes that $cov(b, e) = 0$, such that $y \sim N(X\beta, V)$ with $V = \sigma_e^2(ZD(\theta)Z^T + I)$, where σ_e^2 is the residual variance and I is an identity matrix. Pinheiro and Bates (2000) presented a form that is more convenient for expressing the relative variance matrix D using a relative precision factor Δ, which represents any matrix that satisfies $D^{-1} = \Delta^T\Delta$; Δ is the Cholesky factor of D^{-1}.

The estimation of model parameters using the lmekin function follows the routines implemented in the *lme* function, where the profiled log-likelihood function with respect to θ is calculated using orthogonal-triangular decomposition methods for solving augmented least-squares problems of the form (Pinheiro & Bates, 2000):

$$\begin{bmatrix} Z & X & y \\ \Delta & 0 & 0 \end{bmatrix} = Q_{(i)} \begin{bmatrix} R_{11(i)} & R_{10(i)} & c_{1(i)} \\ 0 & R_{00(i)} & c_{0(i)} \end{bmatrix} \quad (2)$$

where Q is an orthogonal matrix and R is an upper-triangular matrix. The approach of changing the contribution of the marginal distribution of the random effects into extra rows for the response and incidence matrices (left side of the equality above) is called pseudo-data representation because it creates the effect of the marginal distribution by adding "pseudo" observations. To estimate variance components, the function implements a hybrid optimization scheme. It begins by calculating the initial estimates of the θ parameters, then uses a moderate number of EM iterations and switches to Newton-Raphson iterations (Pinheiro & Bates, 2000; Therneau, 2020).

The model in (1) can be employed in the context of GS, where the random effects correspond to the genomic breeding values (g), with $g \sim N(0, \sigma_g^2 K)$; where σ_g^2 is the additive genetic variance, and K is the additive relationship matrix derived from markers (GBLUP). In the case of matrix K derived from the pedigree, the model is the traditional BLUP (Caamal-Pat et al., 2021).

The fit of the GBLUP model using the lmekin function of the coxme package was compared with that of other R packages for GS: rrBLUP (Endelman, 2011), BGLR (Pérez & de los Campos, 2014), sommer (Covarrubias-Pazaran, 2016; Covarrubias-Pazaran, 2018), lme4qtl (Ziyatdinov et al., 2018) and lme4GS (Caamal-Pat et al., 2021). Except for BGLR, all other packages are based on a frequentist approach, where the variance components are estimated by maximum likelihood (ML) or restricted maximum likelihood (REML). The difference between the methods is that REML produces estimates of the variance components that are unbiased because it removes the fixed effects from the model before estimating the variance components. By comparison, in the ML method, the fixed effects are estimated without considering the loss of degrees of freedom due to the estimation of these effects, thus causing bias.

In the BGLR package, the GBLUP model is fitted via Bayesian inference using the Gibbs sampler and the Reproducing Kernel Hilbert space [RKHS] kernel to specify the variance-covariance structures of the genomic breeding values (Pérez & de los Campos, 2014). We considered 30,000 iterations for the Gibbs sampler, with the first 5,000 iterations discarded as burn-in (Caamal-Pat et al., 2021; Crossa et al., 2010). The correct number of iterations and burn-in length are defined when evaluating the convergence of the chains; however, this is beyond the scope of this work. The BGLR assigns scaled-inverse chi-square densities to the variance components, which are indexed by a scale and degree-of-freedom parameter. By default, the degrees of freedom were set equal to five, and the scale parameter was based on the sample variance of the phenotypes. Further details are available in Pérez and de los Campos (2014). In the frequentist analysis, models were fitted using the default parameters of each package.

Applications

To compare the fit of the GBLUP model using different R packages, we used two datasets: a wheat dataset (commonly used in R packages for GS), and a larger dataset simulated by Covarrubias-Pazaran (2016). The wheat dataset consists of a population of 599 wheat lines genotyped using 1447 Diversity Array Technology (DArT) markers, coded as 0 (absence) and 1 (presence). Markers with a minor allele frequency lower than 5% were removed, and missing genotypes were imputed as $x_{ij} \sim Bernoulli(\hat{p}_j)$, where $\hat{p}_j$ is the estimated allele frequency computed from the non-missing genotypes. Following this process, 1279 markers were retained for analysis. The phenotypic trait was the grain yield (2-year average) of the 599 wheat lines evaluated in four environments (Crossa et al., 2010; Pérez & de los Campos, 2014). For this study, we considered only the phenotypes for environment one (low rainfall and irrigation), which were standardized to a sample variance equal to one. Data with phenotypic values (Y), the matrix of markers (X), and the additive relationship matrix computed from a pedigree (A) are publicly available with the BGLR package. The GBLUP model was fitted as follows (VanRaden, 2008; Yang et al., 2010; Mrode, 2014; Resende et al., 2014):

$$y = 1\mu + Zu + e \quad (3)$$

where y is the response vector for the grain yield variable (environment 1), 1 is a vector of ones, and μ is an intercept. Z is an incidence matrix for the genomic breeding values (u), assumed as $u \sim N(0, \sigma_u^2 G)$, where σ_u^2 is the additive genetic variance associated with the markers. G is the genomic relationship matrix, given by $G=XX^T/m$, where X is the matrix of markers centered and standardized (i.e., each marker was centered by subtracting the mean and standardized by dividing by the sample standard deviation), and m is the number of markers (Caamal-Pat et al., 2021). The random error vector was assumed to be in Model (1). To ensure that G is a positive definite matrix, the same can be obtained by $G^* = G + 10^{-6}I$, where I is an identity matrix (Resende et al., 2014). Heritability, given by $h^2 = \sigma_u^2/(\sigma_u^2 + \sigma_e^2)$, measures the extent to which phenotypic variance is explained by additive genetic effects.

Model (3) can be extended by jointly considering the marker and pedigree information. Thus, the following model can be fitted as follows (Crossa et al., 2010; Mrode, 2014; Resende et al., 2014; Caamal-Pat et al., 2021):

$$y = 1\mu + Zu + Ta + e \quad (4)$$

where T is the incidence matrix for additive polygenic effects (a), assumed as $a \sim N(0, \sigma_a^2 A)$; σ_a^2 is the additive genetic variance associated with the pedigree, and A is an additive relationship matrix derived from the pedigree, whose entries are equal to twice the kinship coefficient between pairs of lines (Crossa et al., 2010). The other terms of the model in (4) are the same as those in (3), and it is assumed that the random effects of the model (u, a, and e) are independent. Heritability was then calculated as $h^2 = \sigma_g^2/(\sigma_g^2 + \sigma_e^2)$, with $\sigma_g^2 = (\sigma_u^2 + \sigma_a^2)$. Unlike other R packages considered in the analysis, the lmekin function fits linear mixed models with multiple genetic effects as a linear combination of variance components (Zhao & Luan, 2012; Therneau, 2020). Thus, the function fits the model in (4) as follows:

$$y = 1\mu + Wg + e \quad (5)$$

where $Wg = Zu + Ta$ and W is the incidence matrix for the total genetic effects ($g = u + a$), assumed to be $g \sim N(0, \Sigma)$, where $\Sigma = \sigma_u^2 G + \sigma_a^2 A$, with σ_u^2, σ_a^2, G and A previously defined for models (3) and (4). The equivalence between the models in (4) and (5) results from the assumption of independence between genetic effects. The individual components of g can be obtained as $\hat{u} = \sigma_u^2 G \Sigma^{-1} \hat{g}$ and $\hat{a} = \sigma_a^2 A \Sigma^{-1} \hat{g}$ (Mrode, 2014).

The REML estimates of variance components and the BLUP values obtained using the lmekin function were compared with the values obtained using the other R packages for both genomic models with and without pedigree information. The computing times required to fit the models using the different packages were also recorded.

We additionally implemented a k-fold cross-validation scheme based on the *ranefUvcovNew* function of the lme4GS package (Caamal-Pat et al., 2021) and incorporating the predicted values using the lmekin function. To verify that the cross-validation scheme was correctly employed, the results were compared to those obtained using the *ranefUvcovNew* function in the lme4GS package. The wheat dataset was divided into five disjoint groups (k = 5, five-fold) of approximately equal size. Each of these fifths acted as a validation set and the remainder acted as a training set. GBLUP models with and without pedigree information were fitted using the training set, and the phenotypes for the validation set were predicted. The Pearson correlation between the observed and predicted values for individuals in the validation set and the mean squared error (MSE) were measured. This procedure was repeated five times for each model (with and without pedigree information). Each time, a different group of observations was treated as a validation set, resulting in five estimates of correlation and MSE. The accuracies of the prediction and MSE were computed by averaging these values. In cross-validation, the number of folds is usually fixed at 5 or 10 (James, Witten, Hastie, & Tibshirani, 2013; Caamal-Pat et al., 2021).

In the comparison among R packages, we also considered a big data scenario with phenotypic and genotypic data simulated by Covarrubias-Pazaran (2016) for 5,000 individuals with 10,000 markers for a single trait and a single component of variance, with heritability h^2 = 0,5. The fitted model was the same as in (3), whose genomic relationship matrix was calculated using the A.mat function of the rrBLUP package, and the variance components were estimated by REML. Thus, we compared the computing time of the lmekin function with the other R packages to fit the GBLUP model in a big data scenario. All analyses were performed using R software, version 4.1.3 (R Core Team, 2022), and a PC with a 3.6 GHz Intel Core i3 processor and 8 GB RAM memory. The R codes used to recreate the analyses are available upon request.

Results and discussion

Example 1: Wheat dataset (Markers and Markers + Pedigree)

The results of the BLUPs values, variance component estimates, and computational times using six different R packages are presented in Tables 1 and 2 for the entire dataset. In both fitted models (only markers and markers + pedigree), the estimates obtained using the lmekin function were identical to those obtained using the other R-packages with the frequentist approach (rrBLUP, sommer, lme4qtl, and lme4GS), as shown in Tables 1 and 2. Thus, the lmekin function can also be used in genomic prediction studies.

Table 1. Summary measures of the BLUP values, variance component estimates, and computational times using six different R packages for a model with only marker information.

Estimates	R packages and lmekin function					
	rrBLUP	BGLR	Sommer	lme4qtl	lme4GS	lmekin
Minimum	-2.2668	-2.2719	-2.2668	-2.2668	-2.2668	-2.2668
First quartile	-0.3286	-0.3263	-0.3286	-0.3286	-0.3286	-0.3286
Median	0.1151	0.1108	0.1151	0.1151	0.1151	0.1151
Mean	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Third quartile	0.3918	0.3927	0.3918	0.3918	0.3918	0.3918
Maximum	1.3122	1.3259	1.3122	1.3122	1.3122	1.3122
Additive variance	0.5296	0.5423	0.5296	0.5296	0.5296	0.5296
Residual variance	0.5320	0.5350	0.5320	0.5320	0.5320	0.5320
Time (seconds)	1.69	57.15	4.43	5.82	6.00	12.53

Table 2. Summary measures of the BLUP values, variance component estimates, and computational times using five different R packages for the model with marker and pedigree information.

Estimates	R packages and lmekin function				
	BGLR	sommer	lme4qtl	lme4GS	lmekin
Minimum	-2.2410	-2.2559	-2.2559	-2.2559	-2.2559
First quartile	-0.3797	-0.3806	-0.3806	-0.3806	-0.3806
Median	0.0923	0.0981	0.0981	0.0981	0.0981
Mean	-0.0005	0.0000	0.0000	0.0000	0.0000
Third quartile	0.4536	0.4498	0.4498	0.4498	0.4498
Maximum	1.4101	1.3861	1.3861	1.3861	1.3861
Additive variance (markers)	0.4325	0.4471	0.4471	0.4471	0.4471
Additive variance (pedigree)	0.2343	0.2026	0.2026	0.2026	0.2026
Residual variance	0.4353	0.4338	0.4338	0.4338	0.4338
Time (seconds)	102.35	5.47	39.78	40.02	22.37

Slight differences among the estimates obtained using the BGLR package and those obtained by the other frequentist packages were expected. This is because the GBLUP model is fitted in the BGLR package via Bayesian inference using Gibbs sampling (Pérez & de los Campos, 2014) whereas the other packages fit the model via frequentist inference using the G-REML/G-BLUP procedure. In Bayesian analysis, a higher number of iterations is required to achieve the same parameters than in the frequentist approach.

The rrBLUP package fit the model with only one random effect (marker) in the shortest time (1.69 s), followed by the sommer (4.43 s), and the longest was the BGLR package (57.15 s). The lme4qtl and lme4GS packages, both extensions of the lme4 package (Bates et al., 2015), presented similar computational times (5.82 s and 6.00 s, respectively). The lmekin function, an extension of the lme function (Therneau, 2020), when compared to the lme4 counterpart packages (lme4qtl and lme4GS), took twice as long to fit the same model (Table 1).

For the model with two genetic random effects (Table 2), the lmekin function (22.37 s) was faster than the lme4qtl (39.78 s) and lme4GS (40.02 s) packages, fitting the model in approximately half the time. In the presence of genetic random effects with more than one covariance matrix, the lmekin function differs from other R packages in that it provides a single vector of BLUPs as given by the sum of the genetic effects (Therneau, 2020).

The sommer package had the lowest computational time (only 5.47 s) among the employed packages, at four times faster than the lmekin function, seven times faster than the lme4qtl and lme4GS packages, and 18 times faster than the BGLR (102.35 s). A similar result was found by Caamal-Pat et al. (2021), who compared the computational times of the lme4GS, sommer, and BGLR packages. The speed of sommer is due to the methods implemented in the package to estimate variance components, which combine the direct-inversion Newton–Raphson (NR) or Average Information (AI) algorithms with the auto-decomposition technique of the relationship matrix. The rrBLUP package was not employed because it is limited to a single variance component, in addition to the error (Covarrubias-Pazaran, 2016; Caamal-Pat et al., 2021).

The inclusion of pedigree information reduced the residual variance from 0.53 to 0.43 in the frequentist approach (sommer, lme4qtl, and lme4GS packages and the lmekin function) and from 0.54 to 0.44 in the bayesian approach (BGLR package) (Tables 1 and 2). Once the phenotypes are standardized, the residual variance estimate (σ_e^2) can be used to assess the goodness of fit of the models because it indicates the extent to which the phenotypic variance is not explained by the model (Crossa et al., 2010). Consequently, the

heritability increased from 0.5 (model with only marker information) to 0.6 (model with marker and pedigree information), with approximately 69 and 65% of the total genetic variance being explained by the markers in the frequentist and bayesian approaches, respectively.

Example 2: Five-fold cross-validation using the wheat dataset

The prediction accuracy values obtained via cross-validation using the lmekin function were identical to those obtained using the *ranefUvcovNew* function of the lme4GS package, indicating that the cross-validation scheme was correctly employed. Computing times for cross-validation were also close (116.16 seconds for lme4GS and 109.66 seconds for lmekin in the model including marker and pedigree information). The inclusion of pedigree information in the model increased the prediction accuracy of genomic breeding values (Table 3). Lower mean square error values were also observed for the model based on the markers and pedigree. These results are in agreement with those obtained by de los Campos et al. (2009) and Crossa et al. (2010) for the same dataset.

Table 3. Results of cross-validation (5-fold) using the lmekin function for a set of 599 wheat lines considering only markers and marker + pedigree information.

Fold	Markers		Markers + Pedigree	
	R	MSE	*r*	MSE
1	0.5294	0.7062	0.5621	0.6706
2	0.5810	0.6668	0.6121	0.6296
3	0.4958	0.7907	0.5214	0.7639
4	0.4169	0.7914	0.4428	0.7704
5	0.5947	0.6475	0.6156	0.6278
avg	0.5236	0.7205	0.5508	0.6925
sd	0.0717	0.0678	0.0718	0.0703

MSE: mean square error; avg: average; sd: standard deviation.

The inclusion of pedigree information in genomic prediction models is advantageous in situations with a low number of markers (the case of the 599 wheat lines that were genotyped using only 1279 DART markers). This is because, as the number of markers increases, the relative contribution of the pedigree information tends to decrease (Crossa et al., 2010; de los Campos et al., 2013). The reason for including pedigree information was to show that the lmekin function could be employed to fit GBLUP-type models incorporating multiple genetic effects such as additive, dominance, and epistatic, based on marker and pedigree information.

Example 3: Big data

A dataset simulated by Covarrubias-Pazaran (2016) with 5,000 individuals genotyped with 10,000 markers was used to compare the performance of the lmekin function with its counterparts in a big data scenario with a single random effect. We did not consider big data with more than one random effect owing to the unavailability of a PC, as it would require a system with at least 16 GB of RAM memory (Covarrubias-Pazaran, 2016). The results of the computational time are shown in Figure 1.

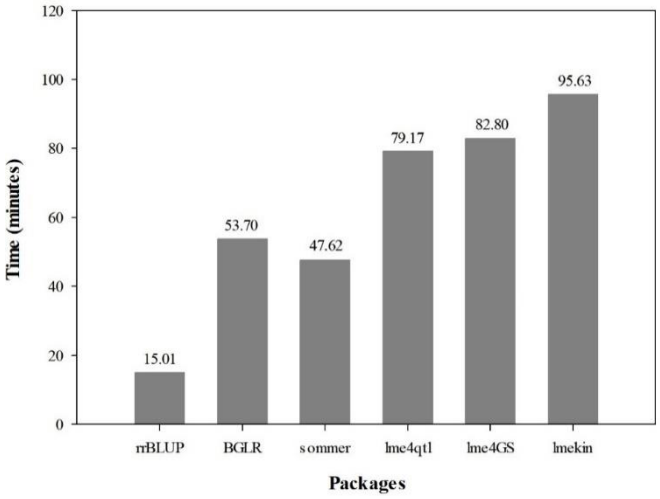

Figure 1. Elapsed times to fit the GBLUP model using different R packages for a simulated dataset with 5,000 individuals and a genetic random effect.

The lmekin function (95.63 min.) as well as the lme4qtl (79.17 min.) and lme4GS (82.8 min.) packages presented the longest execution times, with the lmekin function being approximately 21 and 15% slower than the respective packages. Similar to the wheat dataset, the lme4qtl and lme4GS packages performed similarly. All three programs (lmekin, lme4qtl, and lme4GS) are based on sparse matrix methods, resulting in longer computation times because the genomic relationship matrix is not necessarily sparse. Thus, such packages are expected to have superior computational performance for data with a sparse structure of covariance matrices, such as family based studies (Ziyatdinov et al., 2018).

Among the R-packages compared, rrBLUP presented the lowest computational time (15.01 min.), followed by sommer (47.62 min.), and BGLR (53.7 min), with the last two presenting similar times. These results differed from those obtained by Covarrubias-Pazaran (2016), in that when comparing the packages sommer, ASReml, rrBLUP, regress, BGLR, MCMCglmm, and EMREML, less computational time elapsed using the package sommer for this same dataset. These differences occurred because of the algorithms used. In this study, we used the default parameters of the package, which is based on the direct-inversion NR algorithm, whereas Covarrubias-Pazaran (2016) used the direct inversion AI algorithm; both algorithms are combined with the eigen-decomposition technique of the genomic relationship matrix (Covarrubias-Pazaran, 2016). The rrBLUP and sommer packages implement numerical methods that improve time efficiency when fitting genomic models. In the rrBLUP package, the algorithm is based on the spectral decomposition of kernel ZKZ^T, where Z is the incidence matrix and K is the covariance matrix for random effects (Endelman, 2011).

The main features of the packages used in this study are presented in Table 4. Similar comparisons were made by Covarrubias-Pazaran (2016) and Ziyatdinov et al. (2018). The lmekin function, like the other R packages, excepting the rrBLUP package, allows for predictions of multiple random effects, such as additive and non-additive effects. Furthermore, the lmekin function supports different structures of residual variance. In the same package (coxme) in which the function is embedded, the coxme function is also available, which allows fitting of models for censored responses (Cox model) in the same way as the linear model, i.e., using a kinship matrix (Therneau, 2020). Another R package that supports censored responses is BGLR, which treats censored data as missing and sampled from truncated normal densities. BGLR also supports categorical responses (binary or ordinal multinomial) as well as the lme4qtl package (Pérez & de los Campos, 2014; Ziyatdinov et al., 2018).

Table 4. Comparisons among R-packages used that implement mixed models with genetic random effects.

Features	coxme (lmekin)	rrBLUP	BGLR	sommer	lme4qtl	lme4GS
More than a covariance matrix	✓	×	✓	✓	✓	✓
Covariance for residuals	✓	×	✓	✓	×	×
Use of sparse methods	✓	×	♯	✓	✓	✓
Supports categorical outcomes	×	×	✓	×	✓	♯
Supports censored outcomes	✓	×	✓	×	×	×
Computes other kernels for GS	×	✓	✓	×	×	✓
Restriction on parameters	×	×	♯	♯	✓	×

♯Information not available

Several other statistical software packages have been developed for GS, including several other packages available in the free software R (Budhlakoti et al., 2022). This is a result of the increasing use of GS in plant and animal breeding programs worldwide, which also motivated the use of the lmekin function as an alternative for genomic prediction. As shown, the function has limitations for large datasets; however, it is reliable in the output of the results and efficient in fitting genomic prediction models with multiple random effects. In addition, the methodology can be extended to the context of survival analysis (Santos et al., 2016), by fitting the mixed-effects Cox model in the same manner as the GBLUP model, thereby replacing the pedigree-based relationship matrix with the genomic relationship matrix.

Conclusion

The lmekin function was effective in genomic predictions incorporating multiple random effects and relatively small sample sizes. Differences in computational times occurred because of the different algorithms implemented in the packages to estimate the variance components. The rrBLUP package was the fastest for the scenario with only one genetic random effect, and the time efficiency of the sommer package was

confirmed for the scenario with more than one genetic random effect. The lme4qtl and lme4GS packages, both extensions of the lme4 package, presented similar computational performance rates.

References

Azevedo, C. F., Nascimento, M., Fontes, V. C., Silva, F. F., Resende, M. D. V. D., & Cruz, C. D. (2019). GenomicLand: Software for genome-wide association studies and genomic prediction. *Acta Scientiarum. Agronomy*, *41*(1), 1-7. DOI: http://doi.org/10.4025/actasciagron.v41i1.45361

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1-48. DOI: http://doi.org/10.18637/jss.v067.i01

Budhlakoti, N., Kushwaha, A. K., Rai, A., Chaturvedi, K. K., Kumar, A., Pradhan, A. K., ... Kumar, D. (2022). Genomic Selection: A Tool for Accelerating the Efficiency of Molecular Breeding for Development of Climate Resilient Crops. *Frontiers in Genetics*, *13*, 1-17. DOI: http://doi.org/10.3389/fgene.2022.832153

Caamal-Pat, D., Pérez-Rodríguez, P., Crossa, J., Velasco-Cruz, C., Pérez-Elizalde, S., & Vázquez-Peña, M. (2021). lme4GS: An R-Package for Genomic Selection. *Frontiers in Genetics*, *12*, 1-12. DOI: http://doi.org/10.3389/fgene.2021.680569

Covarrubias-Pazaran, G. (2016). Genome-assisted prediction of quantitative traits using the R package sommer. *PLoS ONE*, *11*(6), 1-15. DOI: http://doi.org/10.1371/journal.pone.0156744

Covarrubias-Pazaran, G. (2018). Software update: Moving the R package sommer to multivariate mixed models for genome-assisted prediction. *bioRxiv*, 1-14. DOI: http://doi.org/10.1101/354639

Crossa, J., Campos, G. D. L., Pérez, P., Gianola, D., Burgueno, J., Araus, J. L., ... Braun, H. J. (2010). Prediction of genetic values of quantitative traits in plant breeding using pedigree and molecular markers. *Genetics*, *186*(2), 713-724. DOI: http://doi.org/10.1534/genetics.110.118521

Cruz, C. D. (2016). Genes software-extended and integrated with the R, Matlab and Selegen. *Acta Scientiarum. Agronomy*, *38*(4), 547-552. DOI: http://doi.org/10.4025/actasciagron.v38i4.32629

de los Campos, G., Hickey, J. M., Pong-Wong, R., Daetwyler, H. D., & Calus, M. P. (2013). Whole-genome regression and prediction methods applied to plant and animal breeding. *Genetics*, *193*(2), 327-345. DOI: http://doi.org/10.1534/genetics.112.143313

de los Campos, G., Naya, H., Gianola, D., Crossa, J., Legarra, A., Manfredi, E., ... Cotes, J. M. (2009). Predicting quantitative traits with regression models for dense molecular markers and pedigree. *Genetics*, *182*(1), 375-385. DOI: http://doi.org/10.1534/genetics.109.101501

Endelman, J. B. (2011). Ridge regression and other kernels for genomic selection with R package rrBLUP. *The Plant Genome*, *4*(3), 250-255. DOI: http://doi.org/10.3835/plantgenome2011.08.0024

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With Applications in R*. New York, NY: Springer.

Meuwissen, T. H., Hayes, B. J., & Goddard, M. (2001). Prediction of total genetic value using genome-wide dense marker maps. *Genetics*, *157*(4), 1819-1829. DOI: http://doi.org/10.1093/genetics/157.4.1819

Mrode, R. A. (2014). *Linear models for the prediction of animal breeding values* (3rd ed.). Boston, MA: CABI.

Muñoz, P. R., Resende Jr, M. F., Gezan, S. A., Resende, M. D. V., de los Campos, G., Kirst, M., ... Peter, G. F. (2014). Unraveling additive from nonadditive effects using genomic relationship matrices. *Genetics*, *198*(4), 1759-1768. DOI: http://doi.org/10.1534/genetics.114.171322

Pérez, P., & de los Campos, G. (2014). Genome-wide regression and prediction with the BGLR statistical package. *Genetics*, *198*(2), 483-495. DOI: http://doi.org/10.1534/genetics.114.164442

Pinheiro, J. C., & Bates, D. M. (2000). *Mixed-effects models in S and S-Plus.* New York, NY: Springer.

Resende, M. D. V. (2016). Software Selegen-REML/BLUP: a useful tool for plant breeding. *Crop Breeding and Applied Biotechnology*, *16*(4), 330-339. DOI: http://doi.org/10.1590/1984-70332016v16n4a49

Resende, M. D. V., Silva, F. F., & Azevedo, C. F. (2014). *Estatística matemática, biométrica e computacional: Modelos mistos, multivariados, categóricos e generalizados (REML/BLUP), inferência bayesiana, regressão aleatória, seleção genômica, QTL-GWAS, estatística espacial e temporal, competição, sobrevivência*. Viçosa, MG: UFV.

R Core Team. (2022). *R: A language and environment for statistical computing*. Vienna, AT: R Foundation for Statistical Computing. Retrieved on May 22, 2022 from https://www.R-project.org

Santos, V. S., Martins Filho, S., Resende, M. D. V., Azevedo, C. F., Lopes, P. S., Guimarães, S. E. F., & Silva, F. F. (2016). Genomic prediction for additive and dominance effects of censored traits in pigs. *Genetics and Molecular Research*, *15*(4), 1-16. DOI: http://doi.org/10.4238/gmr15048764

Therneau T. M. (2020). *Mixed effects cox models*. *R package version 2.2-16*. Retrieved on May 22, 2022 from https://CRAN.R-project.org/package=coxme

VanRaden, P. M. (2008). Efficient methods to compute genomic predictions. *Journal of Dairy Science*, *91*(11), 4414-4423. DOI: http://doi.org/10.3168/jds.2007-0980

Vazquez, A. I., Bates, D. M., Rosa, G. J. M., Gianola, D., & Weigel, K. A. (2010). An R package for fitting generalized linear mixed models in animal breeding. *Journal of Animal Science*, *88*(2), 497-504. DOI: http://doi.org/10.2527/jas.2009-1952

Yang, J., Benyamin, B., McEvoy, B. P., Gordon, S., Henders, A. K., Nyholt, D. R., ... Visscher, P. M. (2010). Common SNPs explain a large proportion of the heritability for human height. *Nature Genetics*, *42*(7), 565-569. DOI: http://doi.org/10.1038/ng.608

Zhao, J. H., & Luan, J. A. (2012). Mixed modeling with whole genome data. *Journal of Probability and Statistics*, *2012*(1), 1-16. DOI: http://doi.org/10.1155/2012/485174

Ziyatdinov, A., Vázquez-Santiago, M., Brunel, H., Martinez-Perez, A., Aschard, H., & Soria, J. M. (2018). lme4qtl: linear mixed models with flexible covariance structure for genetic studies of related individuals. *BMC Bioinformatics*, *19*(1), 1-5. DOI: http://doi.org/10.1186/s12859-018-2057-x