METHODS
published: 18 June 2021
doi: 10.3389/fgene.2021.680569

lme4GS: An R-Package for Genomic Selection

Diana Caamal-Pat[1], *Paulino Pérez-Rodríguez*[1]*, *José Crossa*[1,2]*, *Ciro Velasco-Cruz*[1], *Sergio Pérez-Elizalde*[1] *and Mario Vázquez-Peña*[3]

[1] Department of Socioeconomics, Statistics, and Informatics, Colegio de Postgraduados, Texcoco, Mexico, [2] Biometrics and Statistics Unit, International Maize and Wheat Improvement Center (CIMMYT), Texcoco, Mexico, [3] Department of Irrigation, Universidad Autónoma Chapingo, Texcoco, Mexico

Genomic selection (GS) is a technology used for genetic improvement, and it has many advantages over phenotype-based selection. There are several statistical models that adequately approach the statistical challenges in GS, such as in linear mixed models (LMMs). An active area of research is the development of software for fitting LMMs mainly used to make genome-based predictions. The lme4 is the standard package for fitting linear and generalized LMMs in the R-package, but its use for genetic analysis is limited because it does not allow the correlation between individuals or groups of individuals to be defined. This article describes the new lme4GS package for R, which is focused on fitting LMMs with covariance structures defined by the user, bandwidth selection, and genomic prediction. The new package is focused on genomic prediction of the models used in GS and can fit LMMs using different variance–covariance matrices. Several examples of GS models are presented using this package as well as the analysis using real data.

Keywords: genomic selection, genomic prediction, linear mixed model, lme4, kernel

OPEN ACCESS

Edited by:
Waseem Hussain,
International Rice Research Institute (IRRI), Philippines

Reviewed by:
Jiabo Wang,
Southwest Minzu University, China
Alexandre Bureau,
Laval University, Canada

Correspondence:
Paulino Pérez-Rodríguez
perpdgo@gmail.com
José Crossa
jcrossa@cgiar.org

Specialty section:
This article was submitted to Statistical Genetics and Methodology, a section of the journal Frontiers in Genetics

Received: *14 March 2021*
Accepted: *19 May 2021*
Published: *18 June 2021*

Citation:
Caamal-Pat D, Pérez-Rodríguez P, Crossa J, Velasco-Cruz C, Pérez-Elizalde S and Vázquez-Peña M (2021) lme4GS: An R-Package for Genomic Selection. Front. Genet. 12:680569. doi: 10.3389/fgene.2021.680569

INTRODUCTION

With the new, low-cost, high-throughput genotyping technologies of the last decade, a breeding selection paradigm called genomic selection (GS) has emerged (Meuwissen et al., 2001). GS combines molecular and phenotypic data to obtain the genomic estimated breeding values (GEBVs) of individuals that have been genotyped but not phenotyped (Bernardo and Yu, 2007; de los Campos et al., 2009; Hayes et al., 2009; VanRaden et al., 2009; Crossa et al., 2010). The main advantages of GS over family-based selection in breeding are that it reduces the cost per cycle and the time required for variety development. However, several factors could impact the accuracy of prediction; they occur at different levels and are influenced by several genetic, environmental, and statistical factors.

Complications arise in GS when determining (i) the size and diversity of the training population, (ii) the relationship between the training and testing sets, (iii) genetic complexity, and (iv) the heritability of the traits to be predicted. Challenges in GS are related to the high dimensionality of marker data, where, the number of markers is much larger than the number of observations, the multi-collinearity among markers, the cryptic interaction between

Abbreviations: BGLR, Bayesian generalized linear regression; BLR, Bayesian linear regression; BLUP, best linear unbiased prediction; GEBV, genomic estimated breeding value; GLMM, generalized linear mixed model; GS, genomic selection; LMM, linear mixed model; REML, restricted maximum likelihood.

markers, the complexity of the trait, sample size, correlation among markers, and the ever-present genotype × environment interaction. These complexities require parametric and semi-parametric statistical models, especially mixed models, Bayesian estimations, and, recently, deep machine learning methods that can deal appropriately with the usually large datasets (Crossa et al., 2017). This has led to computational challenges due to the data size and statistical challenges that include model fitting and parameter optimization. Therefore, the development of complete and simple computer packages to estimate the GEBV of the individuals to be selected under these complex scenarios is crucial for an efficient application of GS.

The first R software (R Core Team, 2021) developed for genome-based prediction was presented by de los Campos et al. (2009). Shortly afterward, Pérez et al. (2010) formally described the Bayesian linear regression (BLR) that allows fitting high-dimensional linear regression models including dense molecular markers, pedigree information, and several other covariates other than markers. The BLR R-package described by Pérez et al. (2010) allows including not only markers but also pedigree data jointly. Furthermore, Pérez et al. (2010) explained the challenges that arise when evaluating genomic-enabled prediction accuracy through random cross-validation (CV), as well as how to select the best choice of hyperparameters for the Bayesian models.

Linear mixed models play a fundamental role in GS and genomic-enabled predictions. This kind of models is widely used for predictions, although other models, such as nonlinear models, neural networks, and other machine learning models, could be used for this purpose. The standard linear mixed model of the form $\mathbf{y} = \mathbf{X}\beta + \mathbf{Z}\mathbf{u} + \mathbf{e}$, where, $\mathbf{y}$ is a response vector of dimension $n \times 1$; $\mathbf{X}$ and $\mathbf{Z}$ are the design matrices for the fixed (β) and genotypic random ($\mathbf{u}$) effects, respectively; and two variance components are estimated $\mathbf{u} \sim MN(\mathbf{0}, \sigma_u^2\mathbf{K})$, with $\mathbf{K}$ being a known semidefinite variance–covariance matrix and $\mathbf{e} \sim MN(\mathbf{0}, \sigma_e^2\mathbf{I})$. In the context of GS, $\mathbf{K}$ could be the additive relationship matrix derived from the coefficient of co-ancestry (numerator relationship matrix $\mathbf{A}$), or it could be the genomic relationship matrix obtained from markers ($\mathbf{G}$). As shown below, there are several alternative ways of expressing the incidence matrix $\mathbf{Z}$ and the vector of random effects $\mathbf{u}$ when using the numerical relationship matrix ($\mathbf{A}$). Bayesian versions of linear regression models have been extensively developed, and their companion software largely distributed and used for research and extended to more complicated cases, for example, the introduction of genotype × environment interaction incorporating pedigree and environmental covariables (Jarquín et al., 2014).

Endelman (2011) developed the rrBLUP R-package, which is able to fit the basic linear mixed model with two variance components (σ_u^2 and σ_e^2) described before with the maximum likelihood or restricted maximum likelihood (REML) methods. As an extra facility, the rrBLUP computes the Gaussian kernel and the exponential kernel that usually account for small cryptic epistatic effects among the markers. The rrBLUP has a CV algorithm to measure the prediction accuracy of the models and shows rapid solutions of the mixed model equations for moderate-to-intermediate data sizes. More specialized computer software, such as the synbreed of Wimmer et al. (2012) and GEMMA of Zhou and Stephens (2012), were later developed.

Although the previously mentioned genomic software programs solve important genomic prediction problems (e.g., prediction in training and testing sets, CV, and estimation of variance parameters), they are separate software pieces without a unified statistical and computing framework. So from the user's perspective, having a single package implementing all the models to be fitted will save data preparation time and data analysis time. Thus, Pérez and de los Campos (2014) extended the original BLR R-package developed by Pérez et al. (2010) to a more general R-package, the Bayesian generalized linear regression (BGLR) that offers users a great variety of genomic models and methods in a unified computing software for data analysis. The BGLR is available at CRAN. The BGLR package includes several Bayesian regression models, including parametric variable selection and shrinkage methods, and semi-parametric procedures [Bayesian reproducing kernel Hilbert space (RKHS) regressions]. Many non-genomic applications are implemented as well, and response traits can be continuous or categorical (binary or ordinal). The Bayesian algorithm is based on a Gibbs sampler with scalar updates implemented in efficient routines written in C programming language. Furthermore, the BGLR is the main machinery for adapting other more complex genomic models, for example, the complex phenomenon of genotype × environment interaction including pedigree and environmental covariables (Jarquín et al., 2014). The BGLR is also used for assessing the marker effect × environment interaction of Lopez-Cruz et al. (2015) and for fitting Bayesian ridge regression and the Bayes B, as shown by Crossa et al. (2017), or for using the threshold model for ordinal data as did Montesinos-López et al. (2016), and for running all the Bayesian alphabet models.

Although linear mixed models are important tools for fitting GS models, Covarrubias-Pazaran (2016) mentioned like that current GS software includes only one random effect; and therefore, using genomic prediction for more complicated situations hybrid prediction using additive, dominance, and epistatic effects is not possible under the available models. The authors proposed likelihood-based software for fitting mixed models with multiple random effects that allow the user to specify the variance–covariance structure of random effects. Covarrubias-Pazaran (2016) presented an R-package called sommer for genomic prediction with three algorithms for estimating variance components: average information, expectation–maximization, and efficient mixed model association. Results from sommer were comparable with those of other software, and sommer was faster than its Bayesian counterparts.

The development of software for fitting linear mixed models is an active area of research. The use of pedigree and genomic-enabled prediction linear mixed models is crucial for advancing the application of genomic-assisted breeding. The lme4 package (Bates et al., 2015) for R (R Core Team, 2021) has efficient functions for analyzing linear mixed models and generalized linear mixed models (GLMMs). Some of the main features of lme4 are that (i) it is efficient for large dataset problems; (ii) it handles any number of grouping factors, nested or cross-classified; and (iii) it can use a combination of sparse and dense matrix representations to facilitate the processing of large datasets at high computational speed.

However, the use of lme4 for genetic analysis has been limited because it does not allow using the correlation between individuals or groups of individuals. When individual lines or animals are related, the marginal likelihood must allow using this covariance between relatives. Vazquez et al. (2010) developed a package called pedigreemm that uses the lme4 but allows for correlations between levels of random effects, such as those due to genetic relationships between relatives expressed as pedigree relationships. The methodology of Vazquez et al. (2010) uses the numerator relationship matrix **A** (a positive-definite matrix) and subjects it to the Cholesky decomposition, where, the Cholesky factor (**L**) can be obtained from the pedigree information.

Based on the above considerations and some limitations in terms of the computing efficiency of some existing genomic-enabled prediction models, in this research, we describe the new lme4GS R-Package that is based on the lme4 software of Bates et al. (2015) that is available in CRAN. The lme4GS is focused on genomic-based prediction of GS and can fit mixed models with several different variance–covariance matrices. The lme4GS introduces fixed and random effects, and associated variance–covariance matrices, from which matrices for fixed and random effects ($\mathbf{X}$, $\mathbf{Z}_1, \ldots, \mathbf{Z}_q$, respectively) are obtained. The original variance–covariance matrices are introduced and transformed by using the Cholesky factorization or the eigenvalue decomposition of variance–covariance matrices and later used for defining the objective function (deviance function). Once the objective function has been defined, the optimization module optimizes the objective function and provides REML estimates of the parameters of interest.

MATERIALS AND METHODS

Consider the linear mixed model:

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

where, $\mathbf{y}$ is a response vector of dimensions $n \times 1$, $\mathbf{X}$ is a matrix of fixed effects of dimensions $n \times p$, β is a vector of fixed effects of dimensions $p \times 1$, $\mathbf{Z}$ is an incidence matrix of dimensions $n \times r$, and $\mathbf{u}$ is a vector of random effects. We assume $\mathbf{u} \sim MN(\mathbf{0}, \sigma_a^2\mathbf{K})$ and $\mathbf{e} \sim MN(\mathbf{0}, \sigma_e^2\mathbf{I})$, with $\mathbf{K}$ a known variance–covariance matrix, and σ_a^2 and σ_e^2 are variance parameters associated with $\mathbf{u}$ and $\mathbf{e}$, respectively; furthermore, we assume that $\mathbf{u}$ and $\mathbf{e}$ are independently distributed. In the case of GS, the variance–covariance matrix can be derived from markers or from pedigree.

The linear mixed model (1) can be rewritten as;

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{Z}^*\mathbf{u}^* + \mathbf{e}, \tag{2}$$

where, $\mathbf{Z}^* = \mathbf{Z}\mathbf{L}$, with $\mathbf{L}$ obtained from the Cholesky factorization of $\mathbf{K}$; alternatively, $\mathbf{Z}^* = \mathbf{Z}\boldsymbol{\Gamma}\boldsymbol{\Lambda}^{1/2}$ with $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ the matrices of eigenvectors and eigenvalues, respectively, obtained from the eigenvalue decomposition of $\mathbf{K}$, and $\mathbf{u}^* \sim MN(\mathbf{0}, \sigma_u^2\mathbf{I})$. Note that $\mathbf{Z}^*\mathbf{u}^*$ has the same distribution as $\mathbf{Z}\mathbf{u}$; that is, $\mathbf{Z}^*\mathbf{u}^* \overset{d}{=} \mathbf{Z}\mathbf{u} \sim \mathrm{MN}(\mathbf{0}, \sigma_a^2\mathbf{Z}\mathbf{K}\mathbf{Z}^{'})$.

Best Linear Unbiased Predictions

Once mixed model (2) is fitted, the conditional means of the random effects can be obtained, that is, $\widehat{\mathbf{u}}^*$.The best linear unbiased predictions (BLUPs) for $\mathbf{u}^*$ are obtained as follows: $\widehat{\mathbf{u}}^* = \widehat{\sigma}_u^2\mathbf{Z}^{*'}\widehat{\mathbf{V}}^{*-1}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$ where, $\widehat{\mathbf{V}}^* = \widehat{\sigma}_u^2\mathbf{Z}^*\mathbf{Z}^{*'} + \widehat{\sigma}_e^2\mathbf{I}$, with $\widehat{\sigma}_e^2$, $\widehat{\sigma}_u^2$ and $\widehat{\beta}$ REML estimates of variance parameters and vector of fixed effects, respectively. The conditional means of random effects for the model in equation (1) are obtained as follows: $\widehat{\mathbf{u}} = \mathbf{L}\widehat{\mathbf{u}}^*$ if the Cholesky factorization is used, or alternatively, $\widehat{\mathbf{u}} = \boldsymbol{\Gamma}\boldsymbol{\Lambda}^{1/2}\widehat{\mathbf{u}}^*$ if the eigenvalue is used.

Prediction of New Observations

The main goal of GS is to predict new observations (phenotypic values) or simply obtain the BLUPs for random effects not present in the observed data but drawn from the same population as $\mathbf{u}$ and $\mathbf{e}$ (Gilmour et al., 2004). Assume that the random vector $\mathbf{u}$ and matrix $\mathbf{K}$ are partitioned as follows:

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}, \mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix},$$

the BLUPs for $\mathbf{u}_2$ are obtained as:

$$E\left(\mathbf{u}_2 \mid \mathbf{y}_1\right) = \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{u}_1. \tag{3}$$

In a more general case, model (1) can be extended to include more random effects, that is:

$$\mathbf{y} = \mathbf{X}\beta + \sum_{j=1}^{q} \mathbf{Z}_j\mathbf{u}_j + \mathbf{e}, \tag{4}$$

where, $\mathbf{Z}_j$ is a design matrix of random effects, and $\mathbf{u}_j$ is a vector of random effects, $j = 1, \ldots, q$, where, q corresponds to the number of random terms included in the model. We assume that $\mathbf{u}_j \sim MN(\mathbf{0}, \sigma_j^2\mathbf{K}_j)$ is independently distributed. Note that model (1) is a special case of model (4) obtained by setting $q = 1$, $\mathbf{Z} = \mathbf{Z}_1$, $\mathbf{u} = \mathbf{u}_1, \mathbf{K} = \mathbf{K}_1$, $\sigma_a^2 = \sigma_1^2$. Based on the same computational strategy used to rewrite model (1) as the model in (2), model (4) can be rewritten as:

$$\mathbf{y} = \mathbf{X}\beta + \sum_{j=1}^{q} \mathbf{Z}_j^* u_j^* + \mathbf{e}. \tag{5}$$

Implementation

The lme4GS package is an extension of the lme4 R-package (Bates et al., 2015); lme4GS development was inspired by existing R-packages, pedigreemm (Vazquez et al., 2010) and lme4qtl (Ziyatdinov et al., 2018), which are focused on quantitative trait locus (QTL) mapping association and linkage studies, whereas, lme4GS is focused on the problem of prediction in GS (Meuwissen et al., 2001) with GBLUP-type models, although the models can be applied in other research areas.

lme4GS uses the computational engine provided by the well-tested and widely used lme4 package to fit mixed models with a variance–covariance matrix provided by the user. lme4GS can be considered a generalization of existing package rrBLUP (Endelman, 2011) because it is able to fit model (4), whereas, rrBLUP is able to fit model (1). The package also implements some of the models in the sommer package of Covarrubias-Pazaran (2016). lme4GS uses the high-level modular structure of lmer (formula module, objective function module, optimization module, and output module) to fit the models with variance–covariance matrices provided by the user. The formula module allows the specification of fixed and random effects and associated variance–covariance matrices, from which matrices for fixed and random effects ($\mathbf{X}$, $\mathbf{Z}_1, \dots, \mathbf{Z}_q$, respectively) are obtained. After that, the variance–covariance matrices are introduced by computing transformed incidence matrices ($\mathbf{Z}_j^*, j = 1, \dots, q$) using the Cholesky or eigenvalue decomposition of variance–covariance matrices provided by the user, which are taken as inputs to define the objective function (deviance function). Once the objective function has been defined, the optimization module is used to optimize the objective function and provide REML estimates of the parameters of interest. Finally, the output module is used to provide an output that can be interpreted by the end user. We developed three main R functions:

- **lmerUvcov:** Fits a linear mixed model with a variance–covariance matrix provided by the user. This function takes as input a formula to specify the response $\mathbf{y}$, the fixed effects (fixed) and the random effects (random), a data.frame, and a list (Uvcov) to specify the variance–covariance matrix for random effects. Once the model is fitted, the routine returns an object of class merMod for which many methods are available in R for further processing (e.g., summary, print, predict, and VarCorr).
- **ranefUvcov:** Extracts the conditional means of random effects. This function takes as input an object returned by the lmerUvcov function. If the ranef function in the lme4 package is used taking as input the object provided by the lmerUvcov function, it will extract the conditional means for the random effects in model (6); the conditional means for random effects in model (5) are obtained as explained in the BLUPs section. The ranef function in lme4 is overwritten with ranefUvcov, so the user can call either of these two routines and obtain the same results.
- **ranefUvcovNew:** Obtains BLUPs for new levels of random effects with user-specified variance–covariance matrices. The function takes as input an object provided by the lmerUvcov function and a two-level list with variance–covariance matrices that contains information of the genotype identifiers (GIDs) to be predicted and those that were included when fitting the model. The BLUPs are obtained using partitions similar to those used to derive equation (4).

The software is available in the github repository[1].

BOX 1 | Loading wheat data.
```
library(lme4GS)
library(pedigreemm)
data(wheat599)
ls()   #list objects
```

EXAMPLES

In this section, we illustrate the use of the R-package lme4GS with several examples using sample data included in the package. In our examples, we consider only the prediction of random effects and the estimation of variance parameters, although the package is also able to estimate fixed effects.

Example 1: Genome-Wide Prediction Using Markers and Pedigree

In this example, we analyze a set of 599 wheat lines developed by the CIMMYT Global Wheat Breeding Program. The dataset has

[1]https://github.com/perpdgo/lme4GS

BOX 2 | Computing A and G matrices.
```
## Complete and sort incomplete Pedigree using
  editPed
PedEdit< editPed(sire = wheat.Pedigree$gpid1,
  dam=wheat.Pedigree$gpid2,
                  label = wheat.Pedigree$progenie,
                 verbose = TRUE)

## Converted the data frame PedEdit into an S4 object
of formal
## class ‘Pedigree’
PedFinal<-with(PedEdit,pedigree(label=label,
  sire=sire,dam=dam))

#A
AFull<-getA(PedFinal)
GID<-unique(wheat.Pheno$GID)
selected<-rownames(AFull)%in%GID
A<-AFull[selected,selected]
A<-matrix(A,599,599)
rownames(A)<-colnames(A)<-rownames(AFull
  [selected,selected])

W<-scale(wheat.X,center=TRUE,scale=TRUE)
G<-tcrossprod(W)/ncol(W)

#Environment 1
e1<-which(wheat.Pheno$Env==1)
y<-wheat.Pheno[e1,]$Yield
GID<-as.character(wheat.Pheno[e1,]$GID)

wheat<-data.frame(y = y,mrk=GID,ped=GID)
random<-list(mrk=list(K=G),ped=list(K=A))
fmGA<-lmerUvcov(y∼(1| mrk)+(1|
  ped),data = wheat,Uvcov = random)
summary(fmGA)

#BLUPs
ranefUvcov(fmGA)

#or equivalently
ranef(fmGA)
```

been analyzed several times in the literature (e.g., de los Campos et al., 2009; Crossa et al., 2010; Pérez et al., 2010). The dataset includes grain yield information, a pedigree, and 1,477 markers generated by Triticarte Pty., Ltd. (Canberra, Australia[2]). Here, we present the raw phenotypic data, including the replicates in each environment and the pedigree information, in order to show how to use R tools to obtain the additive relationship matrix that is later used as input for fitting the models. The dataset is loaded into the R environment with the commands shown in **Box 1**.

Once the commands are executed, the following objects are available:

- wheat.Pheno: A data.frame with four columns: Env for environments, Rep for replicates, GID for genotype identifiers, and Yield for grain yield.
- wheat.Pedigree: A data.frame with three columns: gpid1 and gpid2, which correspond to the GID of parents 1 and 2, respectively, and progeny, which correspond to the GIDs of progeny.
- wheat.X: A matrix of dimensions 599 × 1,279, which corresponds to Diversity Array Technology (DArT) markers coded as 0 and 1.

A linear model to predict grain yield in one of the environments using markers and pedigree is given by:

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{Z}_1\mathbf{u}_1 + \mathbf{Z}_2\mathbf{u}_2 + \mathbf{e}, \qquad (6)$$

where, $\mathbf{y}$ is the response vector in one environment, $\mathbf{1}$ is a vector of ones, μ is an intercept, $\mathbf{u}_1 \sim MN(\mathbf{0}, \sigma^2_m\mathbf{G})$, $\mathbf{G} = \mathbf{W}\mathbf{W}'/p$ (see Lopez-Cruz et al., 2015) is a genomic relationship matrix, $\mathbf{W}$ is the matrix of markers centered and standardized, p is the number of markers, σ^2_m is a variance parameter associated with markers, $\mathbf{u_2} \sim MN(\mathbf{0}, \sigma^2_a\mathbf{A})$, $\mathbf{A}$ is an additive relationship matrix derived from pedigree, σ^2_a is its associated variance parameter, $\mathbf{Z}_1$, $\mathbf{Z}_2$ are matrices that connect phenotypes with genotypes, and $\mathbf{e}$ is a random term distributed as in model (1). The additive relationship matrix $\mathbf{A}$ can be easily computed in R using the pedigreemm package (Vazquez et al., 2010); the corresponding Cholesky decomposition can be computed very efficiently, and the package is able to store the result as a sparse matrix. The code in **Box 2** computes the $\mathbf{A}$ and $\mathbf{G}$ matrices and then fits the mixed model using the lmerUvcov function. After that, it extracts the BLUPs using the ranefUvcov function.

The model fitting time is about 81 s on a computer with a 2.8-GHz Intel Core i7 processor. After the model is fitted, the summary function can be used to show some of the results. The estimates of variance parameters are $\widehat{\sigma}^2_m = 0.2218$,

[2]https://www.diversityarrays.com

BOX 3 | Partial output from **Box 2**.

```
Linear mixed model fit by REML ['lmerUvcov']
Formula: y ~ (1 | mrk) + (1 | ped)
   Data: wheat

REML criterion at convergence: 1103.5

Scaled residuals:
      Min       1Q      Median       3Q       Max
 -2.51852  -0.44430    0.00982  0.42390   2.60030

Random effects:
   Groups   Name            Variance    Std.Dev.
   mrk      (Intercept)      0.22189     0.4711
   ped      (Intercept)      0.21138     0.4598
   Residual                  0.03496     0.1870
Number of obs: 1198, groups: mrk, 599; ped, 599

Fixed effects:
              Estimate Std. Error t value
(Intercept)    4.81719   0.08757     55.01
```

BOX 4A | Single training and testing partition.

```
set.seed(456)
trn<-sample(unique(GID),size=as.integer(0.80*599))
tst<-setdiff(unique(GID),trn)

#Phenotypes in training and testing
y_trn<-y[GID%in%trn]
y_tst<-y[GID%in%tst]

A_trn<-A[rownames(A)%in%trn,colnames(A)%in%trn]
G_trn<-G[rownames(G)%in%trn,colnames(G)%in%trn]
GID_trn<-GID[GID%in%trn]
GID_tst<-GID[!(GID%in%trn)]

pheno_trn<-data.frame(y_trn=y_trn,mrk=GID_trn,
                      ped = GID_trn)

random<-list(mrk=list(K=G_trn),ped=list(K=A_trn))

fmGA_trn<-lmerUvcov(y_trn~(1| mrk)+(1| ped),
data=pheno_trn,
                    Uvcov = random)

plot(pheno_trn$y_trn, predict(fmGA_trn),
     xlab="Observed phenotype",ylab="Predicted
     phenotype")

#Predict for new levels
blup_tst<-ranefUvcovNew(fmGA_trn,
                        Uvcov=list(mrk=list(K=G),
                        ped=list(K=A)))
i1<-match(GID_tst,rownames(blup_tst$mrk))
i2<-match(GID_tst,rownames(blup_tst$ped))
blup_mrk<-blup_tst$mrk[i1,1]
blup_ped<-blup_tst$ped[i2,1]
yHat_tst<-fixef(fmGA_trn)[1] + blup_mrk + blup_ped

points(y_tst,yHat_tst,col="red",pch=19)
legend("topleft",legend=c("Training","Testing"),
      pch=c(1,19),col=c("black","red"),bty="n")

#Correlation in testing set
cor(y_tst,yHat_tst)

#MSE
var(y_tst-yHat_tst)

#Data frame with prediction for further processing
predictions<-data.frame(GID=GID_tst,y=y_tst,
yHat=yHat_tst)
```

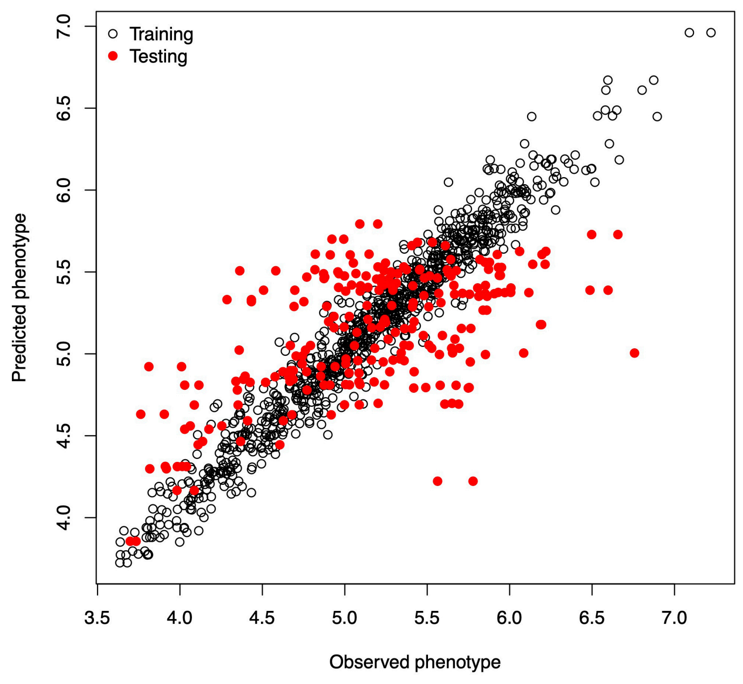

FIGURE 1 | Observed vs. predicted phenotypic values in the training and testing sets.

$\widehat{\sigma}_a^2 = 0.2113$, and $\widehat{\sigma}_e^2 = 0.0349$ (see **Box 3**). The functions predict, residuals, etc., that are routinely used after fitting the model with the lmer function; they can also be used with the resulting object.

Example 2: Training and Testing Sets

In this example, we mimic the GS problem faced by breeders; we evaluate the predictive ability of model (6) by CV, which requires randomly partitioning the data into two disjoint sets, assigning 80% of the lines to the training set and the remaining 20% to the testing set. The code in **Box 4A** partitions the data into the training and testing sets and defines two vectors, y_trn and y_tst, with the phenotypic values of both sets. Next, it creates a list object with the random effects for the linear mixed model. The linear mixed model is fitted using the training set of the data, with the lmerUvcov function. In the next step, we define a list of random effects including the variance–covariance matrices **G** and **A** and the GIDs of the lines to be predicted; the row and column names of the covariance matrices correspond to the GIDs. The ranefUvcovNew function is used for prediction and provides a list of BLUPs for each of the random terms as a result. Finally, the predictions for individuals in the testing set are obtained by simply adding up the intercepts to the BLUPs. Observed and predicted values are stored in a data.frame with three columns: GID, y (observed phenotypic values), and yHat (predicted phenotypic values) used for graphical displays. **Figure 1** shows a scatter plot with observed and predicted phenotypic values in both the training and testing sets. Pearson's correlation coefficient between the observed and predicted values is 0.5638, and the mean squared error (MSE) is 0.2581.

Box 4B shows the R code to perform a five-fold CV that is widely used to study prediction accuracy (e.g., Crossa et al., 2010). We randomly divided the data into five disjoint sets based on the GID, $\{S_1, \ldots, S_5\}$. Each set is used to measure prediction accuracy. With the use of these sets, the data are divided into the training and testing populations; for example, the data in $\{S_2, \ldots, S_5\}$ are the training data, and S_1 are the testing data. The model is fitted using the training data, then phenotypes for S_1 are predicted, and prediction accuracy is measured. The same exercise can be carried out taking S_f as the testing data, $f = 2, \ldots, 5$. **Table 1** shows the results of CV, column 1 corresponds to fold, column 2 shows Pearson's correlation coefficient between observed and predicted values for individuals in the training set, column 3 corresponds to the MSE in the training set, and columns 4 and 5 show the correlations and MSE for individuals in the testing set. The average correlation in the training set is 0.9768, whereas, the correlation in the testing set is 0.5192. The average MSE in the training set is 0.0187, and that in the testing set is 0.2897. The results are as expected: the correlation in the training set is higher than in the testing set, and the MSE is higher in the testing set than in the training set.

Example 3: Hybrid Prediction

The prediction of hybrid performance is very important in agricultural breeding programs. Technow et al. (2014) and Acosta-Pech et al. (2017) employed G-BLUP type models to predict the performance of maize hybrids. The linear model used to that end is given by:

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{W}\theta + \mathbf{Z}_1\mathbf{u}_1 + \mathbf{Z}_2\mathbf{u}_2 + \mathbf{Z}_3\mathbf{u}_3 + \mathbf{e}, \qquad (7)$$

BOX 4B | Cross-validation.

```
set.seed(789)
uGID<-unique(GID)
nFolds<-5
sets<-sample(1:nFolds,size=length(uGID),replace=TRUE)
resultsCV<-matrix(NA,nrow=nFolds,ncol=4)
colnames(resultsCV)=c("r_trn","MSE_trn",
 "r_tst","MSE_tst")

for(f in 1:nFolds)
{
          #Training and testing
          trn<-(uGID[sets!=f])
          tst<-(uGID[sets==f])

          #Phenotypes in training and testing
          y_trn<-y[GID%in%trn]
          y_tst<-y[GID%in%tst]

          A_trn<-A[rownames(A)%in%trn,
          colnames(A)%in%trn]
          G_trn<-G[rownames(G)%in%trn,
          colnames(G)%in%trn]
          GID_trn<-GID[GID%in%trn]
          GID_tst<-GID[!(GID%in%trn)]

          pheno_trn<-data.frame(y_trn=y_trn,
          mrk=GID_trn,
                         ped=GID_trn)

          random<-list(mrk=list(K=G_trn),
          ped=list(K=A_trn))

          fmGA_trn<-lmerUvcov(y_trn~(1| mrk)+(1|
          ped), data=pheno_trn,
                      Uvcov=random)

          yHat_trn<-predict(fmGA_trn)

          #Correlation in training set
          resultsCV[f,1]<-cor(y_trn,yHat_trn)

          #MSE in training set
          resultsCV[f,2]<-var(y_trn-yHat_trn)


          #Predict for new levels
          blup_tst<-ranefUvcovNew(fmGA_trn,
          Uvcov=list(mrk=list(K=G),
          ped=list(K=A)))
          i1<-match(GID_tst,rownames(blup_tst$mrk))
          i2<-match(GID_tst,rownames(blup_tst$ped))
          blup_mrk<-blup_tst$mrk[i1,1]
          blup_ped<-blup_tst$ped[i2,1]
          yHat_tst<-fixef(fmGA_trn)[1] + blup_mrk +
          blup_ped
           #Correlation in testing set
          resultsCV[f,3]<-cor(y_tst,yHat_tst)
          #MSE
          resultsCV[f,4]<-var(y_tst-yHat_tst)
}
resultsCV
```

TABLE 1 | Results from five-fold cross-validation.

Fold	Training		Testing	
	r	**MSE**	***r***	**MSE**
1	0.9752	0.0201	0.5290	0.2778
2	0.9775	0.0181	0.5680	0.2729
3	0.9755	0.0197	0.5096	0.3035
4	0.9786	0.0173	0.4179	0.3280
5	0.9775	0.0182	0.5714	0.2663
avg	0.9769	0.0187	0.5192	0.2897
sd	0.0015	0.0012	0.0624	0.0256

MSE, mean squared error.

BOX 5 | Loading maize data.

```
library(lme4GS)
data(cornHybrids)
ls()  #List objects
```

BOX 6 | Fitting model for hybrid prediction.

```
maize.Pheno$GCA1<-as.character(maize.Pheno$GCA1)
maize.Pheno$GCA2<-as.character(maize.Pheno$GCA2)
maize.Pheno$SCA<-as.character(maize.Pheno$SCA)

#Genomic relationship matrix for parent 1
GCA1<-unique(maize.Pheno$GCA1)
selected<-rownames(maize.G)%in%GCA1
K1<-maize.G[selected,selected]

#Genomic relationship matrix for parent 2
GCA2<-unique(maize.Pheno$GCA2)
selected<-rownames(maize.G)%in%GCA2
K2<-maize.G[selected,selected]

#kronecker, make.dimmanes is necessary to identify
the hybrids
#with the label Parent 1:Parent 2
K3<-kronecker(K1,K2,make.dimnames=TRUE)

#Training set
trn<-which(!is.na(maize.Pheno$PlantHeight))

hybrid<-data.frame(y=maize.Pheno$PlantHeight[trn],
                   loc=maize.Pheno$Location[trn],
                   P1=maize.Pheno$GCA1[trn],
                   P2=maize.Pheno$GCA2[trn],
                   H=maize.Pheno$SCA[trn])

random<-list(P1=list(K=K1),
             P2=list(K=K2),
             H=list(K=K3))

#Fit the model
fm<-lmerUvcov(y~loc+(1|P1)+(1|P2)+(1|H),
 data=hybrid, Uvcov=random)

summary(fm)
```

where, $\mathbf{y}$ is the response vector; $\mathbf{1}$ is a vector of ones; μ is an intercept; $\mathbf{W}$ is the design matrix for environments; θ is the vector of environmental effects (fixed); $\mathbf{Z}_1$, $\mathbf{Z}_2$, and $\mathbf{Z}_3$ are incidence matrices for paternal, maternal, and hybrids, respectively; $\mathbf{u}_1$ and $\mathbf{u}_2$ are vectors of general combining abilities for parental and maternal lines, respectively; $\mathbf{u}_1 \sim MN(\mathbf{0}, \sigma_1^2\mathbf{K}_1)$, $\mathbf{u}_2 \sim MN(\mathbf{0}, \sigma_2^2\mathbf{K}_2)$ with $\mathbf{K}_1$ and $\mathbf{K}_2$ relationship matrices for

BOX 7 | Output from **Box 6**.

```
#...
Random effects:
 Groups  Name         Variance   Std.Dev.
 H      (Intercept)    0.016385   0.12800
 P2     (Intercept)    0.000841   0.02900
 P1     (Intercept)    0.002047   0.04525
 Residual              0.001182   0.03438
Number of obs: 400, groups: H, 100; P2, 20; P1, 20
#...
```

BOX 8 | Predicting hybrid's performance.

```
#Unobserved hybrid performance
blup_tst<-ranefUvcovNew(fm,Uvcov=list(H=list(K=K3)))
blup_tst$H

#variance parameters
vc<-VarCorr(fm)
print(vc,comp=c("Variance","Std.Dev."),digits=4)
variances<-as.data.frame(vc)$vcov
variances

#Heritability
h2<-sum(variances[2:3])/sum(variances[2:4])
h2
```

paternal and maternal lines σ_1^2, σ_2^2 associated variance parameters, $\mathbf{u}_3 \sim MN(\mathbf{0}, \sigma_3^2\mathbf{K}_3)$, with $\mathbf{K}_3 = \mathbf{K}_1 \otimes \mathbf{K}_2$, σ_3^2 variance parameter associated with hybrids, and $\mathbf{e} \sim MN(\mathbf{0}, \sigma_e^2\mathbf{I})$. Note that model (7) can be rewritten as $\mathbf{y} = \mathbf{X}\beta + \mathbf{Z}_1\mathbf{u}_1 + \mathbf{Z}_2\mathbf{u}_2 + \mathbf{Z}_3\mathbf{u}_3 + \mathbf{e}$, where, $\mathbf{X} = [\mathbf{1W}]$ and $\beta = (\mu, \boldsymbol{\theta}')'$, which corresponds to model (4) discussed before. To exemplify how to fit this model in the lme4GS package, we used the DT_cornHybrids dataset included in the R-package sommer (Covarrubias-Pazaran, 2016), and we included a copy of the original data in the package (cornHybrids). The dataset contains phenotypic data for grain yield and plant height for 100 out of 400 possible crosses that originated from 40 inbred lines belonging to two heterotic groups, with 20 lines in each. Only 100 hybrids were evaluated in four locations, and then the problem was to estimate their general combining abilities and specific combining abilities and to predict the performance of untested hybrids at each location. The dataset can be loaded in R using the commands shown in **Box 5**:

The dataset contains the following R objects:

- maize.Pheno: A data.frame with six columns: Location, GCA1 (Parent 1), GCA2 (Parent 2), SCA (hybrid), Yield, and PlantHeight. Records with missing values in the last two columns correspond to hybrids (identified with the Parent 1:Parent 2 label) that were not evaluated in the field and that we need to predict.
- maize.G: A matrix with relationships between individuals for parents of both heterotic groups ($\mathbf{K}_1$ and $\mathbf{K}_2$). The matrix was computed using 511 single-nucleotide polymorphisms (SNPs) using the A.mat function included in the rrBLUP package (Endelman, 2011). The row names and column names of this matrix correspond to the GIDs for Parent 1 and Parent 2.

BOX 9 | Gaussian and exponential kernel.

```
#Box 9: Gaussian and exponential kernel
library(lme4GS)
library(pedigreemm)

#Load data
data(wheat599)

## Complete and sort incomplete Pedigree using editPed
PedEdit<-editPed(sire=wheat.Pedigree$gpid1, dam=wheat.Pedigree$gpid2,
                 label=wheat.Pedigree$progenie, verbose=TRUE)

## Converted the data frame PedEdit into an S4 object of formal
## class 'Pedigree'
PedFinal<-with(PedEdit,pedigree(label=label, sire=sire,dam=dam))

#A
AFull<-getA(PedFinal)
GID<-unique(wheat.Pheno$GID)
selected<-rownames(AFull)%in%GID
A<-AFull[selected,selected]
A<-matrix(A,599,599)
rownames(A)<-colnames(A)<-rownames(AFull[selected,selected])

#X (markers)
X<-scale(wheat.X,center=TRUE,scale=TRUE)

#Phenotypes environment 1
e1<-which(wheat.Pheno$Env==1)
y<-wheat.Pheno[e1,]$Yield
GID<-as.character(wheat.Pheno[e1,]$GID)

wheat<- data.frame(y=y, ped=GID,k_id=GID)

fm1<-theta_optim(y~(1| k_id)+(1| ped), Uvcov=list(ped=list(K=A)),
                 kernel=list(kernel_type="gaussian",MRK=X),
                 data=wheat)

fm2<-theta_optim(y~(1| k_id)+(1| ped), Uvcov=list(ped=list(K=A)),
            kernel=list(kernel_type="exponential",MRK=X),
            data=wheat)

par(mfrow=c(1,2))
plot(fm1$theta,fm1$LL,xlab=expression(theta), ylab="Log-Likelihood",
     type="b",pch=19,main="a)")
plot(fm2$theta,fm2$LL,xlab=expression(theta), ylab="Log-Likelihood",
     type="b",pch=19,main="b)")
```

The code in **Box 6** computes matrices $\mathbf{K}_1$, $\mathbf{K}_2$, and $\mathbf{K}_3$ in model (7) and fits the model using the lmerUvcov function using only observed phenotypic values for plant height.

The model fitting takes about 1 s to complete on a computer with a 2.8-GHz Intel Core i7 processor. Once the model is

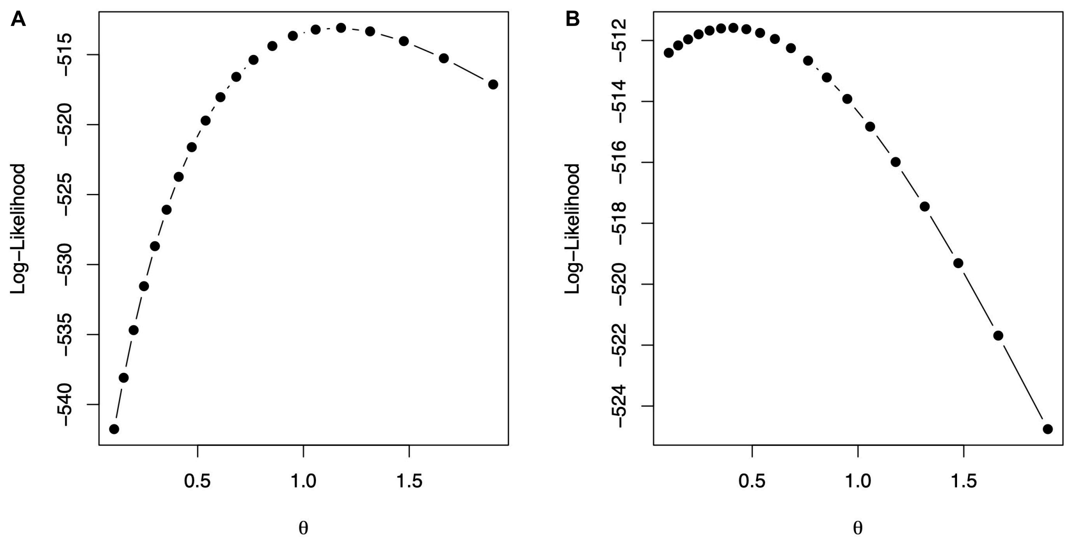

FIGURE 2 | The values of the bandwidth parameter vs. the log-likelihood. **(A)** Gaussian kernel. **(B)** Exponential kernel.

BOX 10 | Summary of fitted models.

```
summary(fm1$fm)
#Output (edited)
Random effects:
 Groups    Name          Variance    Std.Dev.
 k_id     (Intercept)     0.29043     0.5389
 ped      (Intercept)     0.07751     0.2784
 Residual                 0.03434     0.1853
Number of obs: 1198, groups: k_id, 599; ped, 599

Fixed effects:
               Estimate Std. Error t value
(Intercept)     4.6510    0.1314     35.4

summary(fm2$fm)
#Output (edited)
Random effects:
  Groups   Name          Variance    Std.Dev.
  ped     (Intercept)     0.05154     0.2270
  k_id    (Intercept)     0.62952     0.7934
  Residual                0.03408     0.1846
Number of obs: 1198, groups: k_id, 599; ped, 599
Fixed effects:
               Estimate Std. Error t value
(Intercept)     4.5311    0.5599     8.093
```

fitted, the summary function can be used to display some relevant information. The summary output is displayed in **Box 7**, which shows estimates for general combining ability, and specific combining ability and the variance parameter associated with residuals, $\widehat{\sigma}_1^2 = 0.016385, \widehat{\sigma}_2^2 = 0.000841, \widehat{\sigma}_3^2 = 0.002047$, and $\widehat{\sigma}_e^2 = 0.001182$.

The expected hybrid performance of individuals not evaluated in field can be obtained by combining the outputs from the ranefUvcov and ranefUvcovNew functions. **Box 8** shows the instructions to compute the BLUPs for the specific combining ability of hybrids. The ranefUVcov function is called internally in ranefUvcovNew. **Box 8** also shows how to extract variance parameters using the VarCorr function and then compute heritability using the results. Following Covarrubias-Pazaran (2016), $h^2 = (\sigma_1^2 + \sigma_2^2)/(\sigma_1^2 + \sigma_2^2 + \sigma_e^2)$, which leads to an estimated heritability of 0.70.

Example 4: Selection of the Bandwidth Parameter With a Gaussian Kernel

Gianola et al. (2006) introduced the Gaussian kernel into quantitative genetics with the idea of capturing the total genetic effects in the problem of genomic prediction. The Gaussian kernel is defined as (e.g., Morota and Gianola, 2014; Pérez and de los Campos, 2014)

$$\mathbf{K}\left(\mathbf{x}_i,\mathbf{x}_j\right) = \exp\left\{-\theta\frac{d_{ij}^2}{m}\right\} = \exp\left\{-\theta\frac{\sum_{k=1}^{m}\left(x_{ik}-x_{jk}\right)^2}{m}\right\} \quad (8)$$

where, θ is a positive bandwidth parameter; d_{ij} is the Euclidean distance; and $\mathbf{x}_{ik}$ $(i, j = 1, \ldots, n,\ k = 1, \ldots, m)$ is the marker genotype code for individual i at marker k, and m is the number of markers. The bandwidth parameter may be chosen by CV, REML, or maximum likelihood or with Bayesian methods. The Gaussian kernel has been used by many authors for genomic prediction (e.g., de los Campos et al., 2010; Endelman, 2011; Pérez-Elizalde et al., 2015). The selection of the bandwidth is not an easy problem due to high computational cost; de los Campos et al. (2010) and Endelman (2011) proposed evaluating the performance of the model, which includes the Gaussian kernel over a grid of values of θ. Given that $\theta > 0$, if we set $\rho = \exp(-\theta)$, then $\rho \in (0, 1)$, so we can define a grid of values for ρ and then, using these values, set the values for θ, that is, $\theta = -\log\rho$, so that equation (8) can be rewritten as $\mathbf{K}\left(\mathbf{x}_i,\mathbf{x}_j\right) = \exp\left\{\log\rho d_{ij}^2/m\right\}$. Another kernel that is also used in

TABLE 2 | Time comparison (seconds) among different software for models fitted in the work.

Software	Version	Examples			
		Model (6)	Model (7)	Model (10) Gaussian	Model (10) exponential
lme4GS	0.1	81.5	1.3	1,608.8	1,701.1
BGLR 0.8	0.8	143.0	20.2	–	–
sommer 4.1.3	4.1.3	46.0	2.7	–	–

genomic prediction is the exponential kernel (e.g., Piepho, 2009; Endelman, 2011):

$$\mathbf{K}\left(\mathbf{x}_i,\mathbf{x}_j\right) = \exp\left\{-\theta d_{ij}/\sqrt{m}\right\}, \quad (9)$$

where, all the terms have been described previously. Similar to the case of the Gaussian kernel, the model can be reparametrized in terms of parameter $\rho \in (0, 1)$.

We developed the function theta_optim that fits model (5) when one of the random terms ($\mathbf{u}_j, j = 1, ..., q$) includes as the variance–covariance matrix a Gaussian or exponential kernel. This function takes as input the same objects as the lmerUvcov function and a list (kernel) containing (i) a matrix with distances ($\{d_{ij}/\sqrt{m}\}, i, j = 1, ..., n$) or the marker matrix ($\left\{x_{ij}\right\}, i = 1, ..., n, j = 1, ..., m$), (ii) the kernel type (either “gaussian” or “exponential”), and (iii) a sequence of values for θ; the IDs for the individuals are taken directly from the row names of matrices that provide the distances or the markers. If the sequence of values for θ is not provided, then it is generated automatically. The software then fits the mixed model in (5) using the lmerUvcov function for each of the distinct values of θ. The value of θ that maximizes the log-likelihood is chosen as the optimum. The function returns a list with the following elements: a vector of values of the log-likelihood, the maximum value of the log-likelihood, the values of θ used for fitting the model, the optimum value of θ, the fitted model, and the kernel computed with the optimum value of θ .

In the following example, we show how to predict grain yield using a relationship matrix derived from a Gaussian or exponential kernel and a relationship matrix derived from a pedigree. A linear model to predict grain yield for environment one is analogous to model (1):

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{Z}_1\mathbf{u}_1 + \mathbf{Z}_2\mathbf{u}_2 + \mathbf{e}, \quad (10)$$

where, $\mathbf{y}$ is the grain yield; $\mathbf{1}$ is a vector of ones; μ is an intercept, $\mathbf{u}_1 \sim MN(\mathbf{0}, \sigma^2_m\mathbf{K})$, with $\mathbf{K}$ a kernel, which can be either Gaussian or exponential, and σ^2_m is a variance parameter associated with markers; $\mathbf{u_2} \sim MN(\mathbf{0}, \sigma^2_a\mathbf{A})$, where, $\mathbf{A}$ is an additive relationship matrix derived from pedigree, and σ^2_a its associated variance parameter; $\mathbf{Z}_1$, $\mathbf{Z}_2$ are matrices that connect phenotypes with genotypes; and $\mathbf{e}$ is a random term distributed as in model (1).

The code in **Box 9** is used to fit model (10) for Gaussian and exponential kernels. **Figure 2** shows the profile of the log-likelihood for different θ values. For the Gaussian kernel, the maximum of the log-likelihood is equal to -513.0865, attained at $\widehat{\theta} = 1.1779$, whereas, for the exponential kernel, the maximum of the log-likelihood is equal to -511.585, attained at $\widehat{\theta} = 0.4107$. The code in **Box 10** shows how to summarize parameter estimates for the fitted model with the optimum value of the bandwidths from, where, estimates of the variance parameters can be obtained. The model fitting time is about 1,608 s for the model with Gaussian kernel and 1,701 s for the model with exponential kernel using the same processor described before. Note that the selection of bandwidth parameter is a very computer intensive task, but several authors (e.g., Endelman, 2011; Pérez-Elizalde et al., 2015) have reported that the prediction accuracy with nonadditive kernels is higher than the prediction accuracy of ridge regression (or equivalently GBLUP).

Computational Times and Comparison With Other Software

We fitted models (6) and (7) in sommer (Covarrubias-Pazaran, 2016) and BGLR (Pérez and de los Campos, 2014). In the case of BGLR, the number of iterations for the Gibbs sampler was set to 30,000. We were unable to fit the models in rrBLUP (Endelman, 2011) because it is not possible to include more than one covariance matrix in the software; that is also the reason that we were unable to fit model (10) with this software. The predictions from the different software programs were about the same. Here, we present a small comparison of running times for model (6) fitted in **Box 2**, model (7) fitted in **Box 6**, and model (10) fitted in **Box 9** with Gaussian and exponential kernels. Covarrubias-Pazaran (2016) also included a benchmark of sommer against other packages. Models were fitted using a 2.8-GHz Intel Core i7 processor in R-4.0.5 (R Core Team, 2021). **Table 2** presents the resulting time (in seconds) it takes to fit the different models. Some entries in the **Table 2** are empty because the corresponding models cannot be fitted in the corresponding software package. From this **Table 2**, we conclude that sommer is the fastest software, followed by lme4GS and BGLR.

CONCLUSION

We developed an R software package that can be used to fit mixed models with user-defined covariance structures for random effects. The software was developed with applications of GS in mind, mainly for applications in plant breeding with small to moderately sized datasets. However, given the omnipresence of mixed models, the package can be used in other research areas. The software fits the model using well-known and widely tested computational routines available in the lme4 package. The software provides a user-friendly and intuitive interface that allows users to fit a wide variety of classic linear mixed models.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/perpdgo/lme4GS.

AUTHOR CONTRIBUTIONS

DC-P, PP-R, and JC conceived the work and wrote the manuscript. DC-P and PP-R wrote the software. CV-C, SP-E, and MV-P assisted in drafting the manuscript, discussed the analysis, and provided useful comments. All authors contributed to the article and approved the submitted version.

FUNDING

Open-access fees were received from the Bill and Melinda Gates Foundation. We acknowledge the financial support provided by the Bill and Melinda Gates Foundation [INV-003439 BMGF/FCDO Accelerating Genetic Gains in Maize and Wheat for Improved Livelihoods (AG2MW)] as well as USAID projects [Amend. No. 9 MTO 069033, USAID-CIMMYT Wheat/AGGMW, AGG-Maize Supplementary Project, AGG (Stress Tolerant Maize for Africa)]. We are also thankful for the financial support provided by the Foundations for Research Levy on Agricultural Products (FFJ) and the Agricultural Agreement Research Fund (JA) in Norway through NFR grant 267806 as well as the CIMMYT CRP (wheat and maize).

ACKNOWLEDGMENTS

We thank all scientists, field workers, and lab assistants from the National Programs and CIMMYT who collected the data used in this study.

REFERENCES

Acosta-Pech, R., Crossa, J., de los Campos, G., Teyssèdre, S., Claustres, B., Pérez-Elizalde, S., et al. (2017). Genomic models with genotype x environment interaction for predicting hybrid performance: an application in maize hybrids. *Theoretical and Applied Genetics* 130, 1431–1440. doi: 10.1007/s00122-017-2898-0

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

Bernardo, R., and Yu, J. (2007). Prospects for genome wide selection for quantitative traits in maize. *Crop Science* 47, 1082–1090. doi: 10.2135/cropsci2006.11.0690

Covarrubias-Pazaran, G. (2016). Genome-assisted prediction of quantitative traits using the R package sommer. *PloS one* 11:6. doi: 10.1371/journal.pone.0156744

Crossa, J., de los Campos, G., Pérez, P., Gianola, D., Burgueño, J., Araus, J. L., et al. (2010). Prediction of genetic values of quantitative traits in plant breeding using pedigree and molecular markers. *Genetics* 186, 713–724. doi: 10.1534/genetics.110.118521

Crossa, J., Pérez-Rodríguez, P., Cuevas, J., Montesinos-López, O., Jarquín, D., de los Campos, G., et al. (2017). Genomic selection in plant breeding: methods, models, and perspectives. *Trends in plant science* 22, 961–975. doi: 10.1016/j.tplants.2017.08.011

de los Campos, G., Gianola, D., Rosa, G. J., Weigel, K. A., and Crossa, J. (2010). Semi-parametric genomic-enabled prediction of genetic values using reproducing kernel Hilbert spaces methods. *Genetics Research* 92, 295–308. doi: 10.1017/S0016672310000285

de los Campos, G., Naya, H., Gianola, D., Crossa, J., Legarra, A., Manfredi, E., et al. (2009). Predicting quantitative traits with regression models for dense molecular markers and pedigree. *Genetics* 182, 375–385. doi: 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, 250–255. doi: 10.3835/plantgenome2011.08.0024

Gianola, D., Fernando, R. L., and Stella, A. (2006). Genomic-assisted prediction of genetic value with semiparametric procedures. *Genetics* 173, 1761–1776. doi: 10.1534/genetics.105.049510

Gilmour, A., Cullis, B., Welham, S., Gogel, B., and Thompson, R. (2004). An efficient computing strategy for prediction in mixed linear models. *Computational statistics & data analysis* 44, 571–586. doi: 10.1016/S0167-9473(02)00258-X

Hayes, B. J., Bowman, P. J., Chamberlain, A. J., and Goddard, M. E. (2009). Invited review: Genomic selection in dairy cattle: Progress and challenges. *Journal of dairy science* 92, 433–443. doi: 10.3168/jds.2008-1646

Jarquín, D., Crossa, J., Lacaze, X., Du Cheyron, P., Daucourt, J., Lorgeou, J., et al. (2014). A reaction norm model for genomic selection using high-dimensional genomic and environmental data. *Theoretical and applied genetics* 127, 595–607. doi: 10.1007/s00122-013-2243-1

Lopez-Cruz, M., Crossa, J., Bonnett, D., Dreisigacker, S., Poland, J., Jannink, J. L., et al. (2015). Increased prediction accuracy in wheat breeding trials using a marker× environment interaction genomic selection model. *G3: Genes, Genomes, Genetics* 5, 569–582. doi: 10.1534/g3.114.016097

Meuwissen, T. H., Hayes, B. J., and Goddard, M. E. (2001). Prediction of total genetic value using genome-wide dense marker maps. *Genetics* 157, 1819–1829.

Montesinos-López, O. A., Montesinos-López, A., Crossa, J., Toledo, F. H., Pérez-Hernández, O., Eskridge, K. M., et al. (2016). A genomic Bayesian multi-trait and multi-environment model. *G3: Genes, Genomes, Genetics* 6, 2725–2744. doi: 10.1534/g3.116.032359

Morota, G., and Gianola, D. (2014). Kernel-based whole-genome prediction of complex traits: a review. *Frontiers in genetics* 5:363. doi: 10.3389/fgene.2014.00363

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

Pérez, P., de los Campos, G., Crossa, J., and Gianola, D. (2010). Genomic-enabled prediction based on molecular markers and pedigree using the Bayesian linear regression package in R. *The plant genome* 3, 106–116. doi: 10.3835/plantgenome2010.04.0005

Pérez-Elizalde, S., Cuevas, J., Pérez-Rodríguez, P., and Crossa, J. (2015). Selection of the bandwidth parameter in a Bayesian kernel regression model for genomic-enabled prediction. *Journal of agricultural, biological, and environmental statistics* 20, 512–532. doi: 10.1007/s13253-015-0229-y

Piepho, H. P. (2009). Ridge regression and extensions for genomewide selection in maize. *Crop Sci.* 49, 1165–1176. doi: 10.2135/cropsci2008.10.0595

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing.

Technow, F., Schrag, T. A., Schipprack, W., Bauer, E., Simianer, H., and Melchinger, A. E. (2014). Genome properties and prospects of genomic prediction of hybrid performance in a breeding program of maize. *Genetics* 197, 1343–1355. doi: 10.1534/genetics.114.165860

VanRaden, P. M., Van Tassell, C. P., Wiggans, G. R., Sonstegard, T. S., Schnabel, R. D., Taylor, J. F., et al. (2009). Invited review: Reliability of genomic predictions for North American Holstein bulls. *Journal of dairy science* 92, 16–24. doi: 10.3168/jds.2008-1514

Vazquez, A. I., Bates, D. M., Rosa, G. J. M., Gianola, D., and Weigel, K. A. (2010). an R package for fitting generalized linear mixed models in animal breeding. *Journal of animal science* 88, 497–504. doi: 10.2527/jas.2009-1952

Wimmer, V., Albrecht, T., Auinger, H. J., and Schön, C. C. (2012). synbreed: a framework for the analysis of genomic prediction data using R. *Bioinformatics* 28, 2086–2087. doi: 10.1093/bioinformatics/bts335

Zhou, X., and Stephens, M. (2012). Genome-wide efficient mixed-model analysis for association studies. *Nature genetics* 44, 821–824. doi: 10.1038/ng.2310

Ziyatdinov, A., Vázquez-Santiago, M., Brunel, H., Martinez-Perez, A., Aschard, H., and Soria, J. M. (2018). lme4qtl: linear mixed models with flexible covariance structure for genetic studies of related individuals. *BMC bioinformatics* 19:68. doi: 10.1186/s12859-018-2057-x

Conflict of Interest: 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.