Chapter 5
Linear Mixed Models

5.1 General of Linear Mixed Models

Linear mixed models (LMM) are flexible extensions of linear models in which fixed and random effects enter linearly into the model. This is useful in many disciplines to model repeated, longitudinal, or clustered observations, in which random effects are introduced to help capture correlation or/and random variation among observations in the same group of individuals. Random effects are random values associated with levels of a random factor, and often represent random deviations from the population mean and linear relationships described by fixed effects (Pinheiro and Bates 2000; West et al. 2014).

The first formulation of a linear mixed model was applied in the field of astronomy to analyze repeated telescopic observations made at various hourly intervals over a range of nights (West et al. 2014). The mixed model approach is often called by various names, depending on the discipline in which it is applied. For example, in the social sciences, this approach is known as a multilevel or hierarchical model that is often used to flexibly measure the different levels of grouping present in the data structure (e.g., an impact evaluation of a new teaching method, survey of job satisfaction, education applications, etc.) (Goldstein 2011; Speelman et al. 2018; Finch et al. 2019).

Other application areas can be found in medicine (health care research, Leyland and Goldstein 2001; Brown and Prescott 2014), agriculture, ecology, industry, and animal science, where this model is often referred to as the random effects or mixed-effects model (Pinheiro and Bates 2000; Raudenbush and Bryk 2002; Meeker et al. 2011; Zuur et al. 2009). Specifically, now there is an increasing number of applications of this model in genomic selection for plant and animal breeding, where molecular markers obtained by genotyping-by-sequencing or other technologies are used to predict breeding values for non-phenotyped lines to select candidate lines prior to phenotypic evaluation (Meuwissen et al. 2001; Poland et al. 2012; Cabrera-Bosquet et al. 2012; Araus and Cairns 2014; Crossa et al. 2017;

O. A. Montesinos López et al., *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, https://doi.org/10.1007/978-3-030-89010-0_5

Covarrubias-Pazaran et al. 2018; Wang et al. 2018; Cappa et al. 2019). However, the use of this model in animal science can be traced back to Henderson (1950).

The general univariate linear mixed model (Harville 1977) is provided by the formula

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{Zb} + \boldsymbol{\epsilon}, \tag{5.1}$$

where $\boldsymbol{Y}$ is the $n \times 1$ random response vector, $\boldsymbol{X}$ is the $n \times (p + 1)$ design matrix for the fixed effects, $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^{\mathrm{T}}$ is the $(p + 1) \times 1$ coefficient vector of fixed effects, $\boldsymbol{b}$ is a $q \times 1$ vector of random effects, $\boldsymbol{Z}$ is the associate matrix design for the random effects, and $\boldsymbol{\epsilon}$ is the $n \times 1$ vector of random errors. It assumes that $\boldsymbol{\epsilon}$ is a random vector with a mean vector of $\boldsymbol{0}$ and a variance–covariance matrix $\boldsymbol{R}$, $\boldsymbol{b}$ is a random vector with a mean of $\boldsymbol{0}$ and variance–covariance matrix $\boldsymbol{D}$, and a null variance–covariance matrix between $\boldsymbol{\epsilon}$ and $\boldsymbol{b}$, $\mathrm{Cov}(\boldsymbol{\epsilon}, \boldsymbol{b}) = \boldsymbol{0}_{n \times q}$. In genomic applications, $\boldsymbol{b}$ often includes the genotypic effects and genotype × environment interaction effects, while $\boldsymbol{X}$ may contain information about environment covariates and other related information.

Note that under this model, $E(\boldsymbol{Y}) = \boldsymbol{X\beta}$ and the variance–covariance matrix of the response vector is $\mathrm{Var}(\boldsymbol{Y}) = \boldsymbol{ZDZ}^{\mathrm{T}} + \boldsymbol{R}$.

5.2 Estimation of the Linear Mixed Model

5.2.1 Maximum Likelihood Estimation

One method typically used for the estimation of the parameters of the LMM is the maximum likelihood approach. For the estimation under an LMM, the random errors and the random effects components are needed. Assuming that $\boldsymbol{\epsilon} \sim N_n(\boldsymbol{0}, \boldsymbol{R})$, and $\boldsymbol{b} \sim N_q(\boldsymbol{0}, \boldsymbol{D})$, with $\boldsymbol{R}$ and $\boldsymbol{D}$ positive semi-defined matrices, the marginal distribution of the response vector $\boldsymbol{Y}$ is $N_n(\boldsymbol{X\beta}, \boldsymbol{ZDZ}^{\mathrm{T}} + \boldsymbol{R})$, and so the likelihood of the parameters is given by

$$L(\boldsymbol{\beta}, \boldsymbol{D}, \boldsymbol{R}; \boldsymbol{y}) = \frac{|\boldsymbol{V}|^{-\frac{1}{2}}}{(2\pi)^{\frac{n}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{X\beta})^{\mathrm{T}} \boldsymbol{V}^{-1} (\boldsymbol{y} - \boldsymbol{X\beta})\right], \tag{5.2}$$

where $\boldsymbol{V} = \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{DZ} + \boldsymbol{R}$ is the marginal variance of $\boldsymbol{Y}$.

The maximum likelihood estimators (MLE) of the parameters, $\boldsymbol{\beta}$, $\boldsymbol{D}$, and $\boldsymbol{R}$, are the values that maximize the likelihood function (5.2) (Searle et al. 2006; Stroup 2012), but due to the fact that no explicit formulas to estimate these parameters exist, numerical methods such as Newton–Raphson and Fisher Scoring are used. See details for implementing these methods in Jennrich and Sampson (1976) for the case where: $\boldsymbol{D}$ is a block diagonal with a submatrix in each diagonal of the form $\sigma_j^2 \boldsymbol{A}_j$, where $\boldsymbol{A}_j$ is a known matrix and σ_j^2 is the variance component parameter to be estimated in this case; while for $\boldsymbol{R} = \sigma^2\boldsymbol{C}$, where $\boldsymbol{C}$ is a known matrix, only σ^2 should be estimated. For a more general explanation, see Jennrich and Schluchter (1986), and for an improvement of the algorithms proposed by these authors, consult Lindstrom and Bates (1988).

Another numerical method that can be used to obtain the MLE is the expected maximization (EM) algorithm, which is conceptually a simple algorithm for parameter estimation in this model (Laird and Ware 1982). This algorithm is an iterative numerical method to obtain maximum likelihood in the context of missing or hidden data (Borman 2004). The EM algorithm is described for the case where $\boldsymbol{R} = \sigma^2\boldsymbol{I}_n$, and where $\boldsymbol{I}_n$ is the identity matrix of dimension n. This algorithm, for some specific variance–covariance matrices of random effects, as described and used below, can be implemented using the sommer R package (Covarrubias-Pazaran 2016, 2018), which provides two additional algorithms available to obtain the MLE of the parameters in this same model.

5.2.1.1 EM Algorithm

The likelihood for complete data, $\boldsymbol{y}$ and $\boldsymbol{b}$, is given by

$$\begin{aligned} f_{\boldsymbol{Y},\boldsymbol{b}}(\boldsymbol{y},\boldsymbol{b}) &= f_{\boldsymbol{Y}|\boldsymbol{b}}(\boldsymbol{y}|\boldsymbol{b})f_{\boldsymbol{b}}(\boldsymbol{b}) \\ &= \frac{|\boldsymbol{D}|^{-\frac{1}{2}}}{(2\pi\sigma^2)^{\frac{n}{2}}}\exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X\beta}-\boldsymbol{Zb})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X\beta}-\boldsymbol{Zb})-\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{D}^{-1}\boldsymbol{b}\right] \end{aligned}$$

As such, the log-likelihood for the complete data, $\boldsymbol{y}$ and $\boldsymbol{b}$, is given by

$$\begin{aligned} \ell_c(\boldsymbol{\beta},\boldsymbol{\theta};\boldsymbol{y},\boldsymbol{b}) = &\ \log\left[f_{\boldsymbol{Y},\boldsymbol{b}}(\boldsymbol{y},\boldsymbol{b})\right] \\ = &-\frac{n}{2}\log\left(2\pi\sigma^2\right)-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X\beta}-\boldsymbol{Zb})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X\beta}-\boldsymbol{Zb}) \\ &-\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{D}^{-1}\boldsymbol{b}-\frac{1}{2}\log\left(|\boldsymbol{D}|\right), \end{aligned}$$

where $\boldsymbol{\theta}$ is the vector parameter that defines the variance–covariance matrix of the random effects ($\boldsymbol{D}$) and the random vector ($\boldsymbol{R}$). Some specific examples are given below.

E Step

Because $E(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}) = \mathrm{tr}[\boldsymbol{A}\mathrm{Var}(\boldsymbol{u})] + E(\boldsymbol{u})^{\mathrm{T}}\boldsymbol{A}E(\boldsymbol{u})$ and $\boldsymbol{b} \mid \boldsymbol{Y} = \boldsymbol{y} \sim N_q\left(\widetilde{\boldsymbol{b}},\widetilde{\boldsymbol{D}}\right)$ (see Appendix 1), given the current values of the parameters $\boldsymbol{\beta}_{(t)}$ and $\boldsymbol{\theta}_{(t)}$, the conditional expected value of the complete likelihood, $\ell_c(\boldsymbol{\beta},\boldsymbol{\theta};\boldsymbol{y},\boldsymbol{b})$, is given by [Step (E)]:

$$
\begin{aligned}
Q\left(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{\beta}_{(t)},\boldsymbol{\theta}_{(t)},\right) &= E_{\boldsymbol{b}|\boldsymbol{y}}[\ell_c(\boldsymbol{\beta},\boldsymbol{b},\boldsymbol{\theta};\boldsymbol{y})] \\
&= -\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\mathrm{tr}\left(\boldsymbol{Z}\widetilde{\boldsymbol{D}}_{(t)}\boldsymbol{Z}^{\mathrm{T}}\right) - \frac{1}{2\sigma^2}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{Z}\widetilde{\boldsymbol{b}}_{(t)}\right)^{\mathrm{T}}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{Z}\widetilde{\boldsymbol{b}}_{(t)}\right) \\
&\quad -\frac{1}{2}\mathrm{tr}\left(\boldsymbol{D}^{-1}\widetilde{\boldsymbol{D}}_{(t)}\right) - \frac{1}{2}\widetilde{\boldsymbol{b}}_{(t)}^{\mathrm{T}}\boldsymbol{D}^{-1}\widetilde{\boldsymbol{b}}_{(t)} - \frac{1}{2}\log\left(|\boldsymbol{D}|\right) \\
&= -\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\left[\mathrm{tr}\left(\boldsymbol{Z}\widetilde{\boldsymbol{D}}_{(t)}\boldsymbol{Z}^{\mathrm{T}}\right) + \left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{Z}\widetilde{\boldsymbol{b}}_{(t)}\right)^{\mathrm{T}}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{Z}\widetilde{\boldsymbol{b}}_{(t)}\right)\right] \\
&\quad -\frac{1}{2}\left\{\mathrm{tr}\left[\left(\widetilde{\boldsymbol{D}}_{(t)}+\widetilde{\boldsymbol{b}}_{(t)}\widetilde{\boldsymbol{b}}_{(t)}^{\mathrm{T}}\right)\boldsymbol{D}^{-1}\right] + \log\left(|\boldsymbol{D}|\right)\right\}
\end{aligned}
$$

where $\widetilde{\boldsymbol{D}}_{(t)} = \left(\boldsymbol{D}_{(t)}^{-1} + \sigma_{(t)}^{-2}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z}\right)^{-1}$, $\widetilde{\boldsymbol{b}}_{(t)} = \sigma_{(t)}^{-2}\widetilde{\boldsymbol{D}}_{(t)}\boldsymbol{Z}^{\mathrm{T}}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_{(t)}\right)$, and $\boldsymbol{\beta}_{(t)}$, $\sigma_{(t)}^2$, and $\boldsymbol{D}_{(t\,+\,1)}$ are the current values of $\boldsymbol{\beta}$, σ^2, and $\boldsymbol{D}$, respectively.

M Step

The second step of the EM algorithm is the M step, which consists of updating the parameters by maximizing the conditional expected value of the complete likelihood. First, we can observe that for any value of $\boldsymbol{\theta}$, the value of $\boldsymbol{\beta}$ that maximizes $Q(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{\beta}_{(t)},\boldsymbol{\theta}_{(t)})$ is given by

$$
\boldsymbol{\beta}_{(t+1)} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\left(\boldsymbol{y}-\boldsymbol{Z}\widetilde{\boldsymbol{b}}_{(t)}\right)
$$

which does not depend on the chosen values of $\boldsymbol{\theta}$, but rather specifically on the chosen values of σ^2 and $\boldsymbol{D}$. Then by equating to zero, the derivative of $Q(\boldsymbol{\beta}_{(t\,+\,1)},\boldsymbol{\theta}|\boldsymbol{\beta}_{(t)},\boldsymbol{\theta}_{(t)})$ with respect to σ^2, and solving for σ^2, we can obtain that the value of σ^2 that maximizes $Q(\boldsymbol{\beta}_{(t\,+\,1)},\boldsymbol{\theta}|\boldsymbol{\beta}_{(t)},\boldsymbol{\theta}_{(t)})$, for $\boldsymbol{D}$ fixed, is given by

$$
\sigma_{(t+1)}^2 = \frac{1}{n}\left[\mathrm{tr}\left(\boldsymbol{Z}\widetilde{\boldsymbol{D}}_{(t)}\boldsymbol{Z}^{\mathrm{T}}\right) + \left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_{(t+1)}-\boldsymbol{Z}\widetilde{\boldsymbol{b}}_{(t)}\right)^{\mathrm{T}}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_{(t+1)}-\boldsymbol{Z}\widetilde{\boldsymbol{b}}_{(t)}\right)\right]
$$

which is independent of the value of $\boldsymbol{D}$. Now, according to result 4.10 in Johnson and Wichern (2002), the value of $\boldsymbol{D}$ that maximizes $Q(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{\beta}_{(t)},\boldsymbol{\theta}_{(t)})$ is given by

$$
\boldsymbol{D}_{(t+1)} = \widetilde{\boldsymbol{D}}_{(t)} + \widetilde{\boldsymbol{b}}_{(t)}\widetilde{\boldsymbol{b}}_{(t)}^{\mathrm{T}}
$$

and does not depend on $\boldsymbol{\beta}$ and σ^2. So, by joining the above optimization, we have the M step that consists of updating the parameters $\boldsymbol{\beta}$, σ^2, and $\mathbf{D}$, with $\boldsymbol{\beta}_{(t\,+\,1)}$, $\sigma_{(t+1)}^2$, and $\boldsymbol{D}_{(t\,+\,1)}$, respectively. At this point, we can observe that the current value of the parameters $\boldsymbol{\beta}_{(t)}$, $\sigma_{(t)}^2$, and $\boldsymbol{D}_{(t)}$ are used in the computation of $\widetilde{\boldsymbol{b}}$ and $\widetilde{\boldsymbol{D}}$.

In the case of $\boldsymbol{D} = \sigma_g^2\boldsymbol{A}$, where $\boldsymbol{A}$ is a known matrix (the case in some genomic prediction models, where $\boldsymbol{A}$ corresponds to the genomic relationship matrix, the pedigree matrix, or the environmental matrix), the unique variance parameters to estimate are σ_g^2 and σ^2, that is, $\left(\boldsymbol{\theta} = \left(\sigma_g^2, \sigma^2\right)\right)$. In the same fashion, very often in genomic applications but also in a more general setting, $\boldsymbol{D} = \mathrm{Diag}\left(\sigma_1^2 \boldsymbol{A}_1, \ldots, \sigma_K^2\, \boldsymbol{A}_K\right)$, where $\boldsymbol{A}_k$ represents a known variance–covariance matrix (or correlation) structure (genomic relationship matrix, pedigree relationship matrix, etc., Burgueño et al. 2012) between the different random effects included. In this context, the variance component parameters to estimate are $\sigma_k^2, k = 1, \ldots, K,$ and σ^2 $\left(\boldsymbol{\theta} = \left(\sigma_1^2, \ldots, \sigma_K^2, \sigma^2\right)\right)$, where $\boldsymbol{A}_k$, $k = 1, \ldots, K$, are positive defined known matrices of dimensions $q_k \times q_k$, $k = 1, \ldots, K$, such that $\sum_{k=1}^{K} q_k = q$, the E step can be reduced (see Appendix 2) to

$$Q\left(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{\beta}_{(t)},\boldsymbol{\theta}_{(t)}\right) = -\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\left[\mathrm{tr}\left(\boldsymbol{Z}\widetilde{\boldsymbol{D}}_{(t)}\boldsymbol{Z}^{\mathrm{T}}\right) + \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\widetilde{\boldsymbol{b}}_{(t)}\right)^{\mathrm{T}}\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\widetilde{\boldsymbol{b}}_{(t)}\right)\right]$$
$$-\frac{1}{2}\sum_{k=1}^{K}\left\{\frac{\sigma_{k(t)}^2}{\sigma_k^2}\left[q_k - \sigma_{k(t)}^2 \mathrm{tr}\left(\boldsymbol{A}_k \boldsymbol{Z}_k^{\mathrm{T}} \boldsymbol{V}_{(t)}^{-1} \boldsymbol{Z}_k\right) + \sigma_{k(t)}^{-2} \widetilde{\boldsymbol{b}}_{k(t)}^{\mathrm{T}} \boldsymbol{A}_k^{-1} \widetilde{\boldsymbol{b}}_{k(t)}\right] + q_k \log\left(\sigma_k^2\right) + \log\left(|\boldsymbol{A}_k|\right)\right\},$$

where $\boldsymbol{V}_{(t)}$ is the marginal matrix of variance–covariance of the response vector in the current value of the parameters, $\boldsymbol{Z} = [\mathbf{Z}_1 \mathbf{Z}_2 \cdots \mathbf{Z}_K]$ is the partitioned design matrix of random effects, with $\boldsymbol{Z}_k$ $n \times q_k$, the corresponding matrix design for the random effects k, $\boldsymbol{b}_k$ $(\boldsymbol{b}^{\mathrm{T}} = \left(\boldsymbol{b}_1^{\mathrm{T}}, \ldots, \boldsymbol{b}_K^{\mathrm{T}}\right))$, $\widetilde{\boldsymbol{b}}_{(t)}^{\mathrm{T}} = \left(\widetilde{\boldsymbol{b}}_{1(t)}^{\mathrm{T}}, \ldots, \widetilde{\boldsymbol{b}}_{K(t)}^{\mathrm{T}}\right)$, and $\sigma_{k(t)}^2, k = 1, \ldots, K,$ are the current values of the variance parameters. Finally, for this specific model, the maximization updates for the beta coefficients and variance components are the same as before, where the variance components are

$$\sigma_{k(t+1)}^2 = \frac{1}{q_k}\sigma_{k(t)}^2\left[q_k - \sigma_{k(t)}^2 \mathrm{tr}\left(\boldsymbol{A}_k \boldsymbol{Z}_k^{\mathrm{T}} \boldsymbol{V}_{(t)}^{-1} \boldsymbol{Z}_k\right) + \sigma_{k(t)}^{-2} \widetilde{\boldsymbol{b}}_{k(t)}^{\mathrm{T}} \boldsymbol{A}_k^{-1} \widetilde{\boldsymbol{b}}_{k(t)}\right], k = 1, \ldots, K.$$

These are obtained by maximizing $Q(\boldsymbol{\beta}, \boldsymbol{\theta}|\, \boldsymbol{\beta}_{(t)}, \boldsymbol{\theta}_{(t)})$ (defined above) with respect to $\sigma_k^2, k = 1, \ldots, K.$

5.2.1.2 REML

An alternative to the ML estimation of the variance components of model (5.1) and to avoid the underestimation of the maximum likelihood method is the restricted maximum likelihood estimation method (REML) proposed by Patterson and Thompson (1971). Among the several ways to define this, one is discussed by Laird and Ware (1982), which under a Bayesian paradigm, consists of estimating the parameters of the variance components by maximizing the marginal posterior distribution of the variance components by assuming a "locally" uniform prior to the distribution for $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, that is, $f(\boldsymbol{\beta}, \boldsymbol{\theta}) \propto 1$ (Pinheiro and Bates 2000). The marginal posterior of the variance components is given in Appendix 3

$$
\begin{aligned}
f(\boldsymbol{\theta}|\boldsymbol{y}) &\propto \int f(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\beta} \propto \int f(\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{\theta})d\boldsymbol{\beta} \\
&\propto |\boldsymbol{V}|^{-\frac{1}{2}}|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}|^{\frac{1}{2}}\exp\left\{-\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{X}\widetilde{\boldsymbol{\beta}}\right)^{\mathrm{T}}\boldsymbol{V}^{-1}\left(\boldsymbol{y}-\boldsymbol{X}\widetilde{\boldsymbol{\beta}}\right)\right\},
\end{aligned}
$$

where $\widetilde{\boldsymbol{\beta}} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{y}$, which corresponds to the maximum likelihood of the fixed effects when the variance components are assumed known as generalized least squares estimates of $\boldsymbol{\beta}$ (GLS). Then, the restricted maximum likelihood estimators (REML) $\boldsymbol{\theta}$ are those that maximize $f(\boldsymbol{\theta}|\,\boldsymbol{y})$, or equivalently, the REML $\boldsymbol{\theta}$ can be defined as maximizing the

$$
\ell_R(\boldsymbol{\theta};\boldsymbol{y}) = -\frac{1}{2}\log\left(\left|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\right|\right) - \frac{1}{2}\log\left(|\boldsymbol{V}|\right) - \frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{X}\widetilde{\boldsymbol{\beta}}\right)^{\mathrm{T}}\boldsymbol{V}^{-1}\left(\boldsymbol{y}-\boldsymbol{X}\widetilde{\boldsymbol{\beta}}\right)
$$

This function is known as the restricted likelihood because it can be shown that this also corresponds to the likelihood associated with the maximum number $(n-p-1)$ of linearly independent error contrasts $\boldsymbol{FY}$, where $\boldsymbol{F}$ is a full row rank $(n-p-1)\times n$ known matrix such that $\boldsymbol{FX}=\boldsymbol{0}$. It is important to point out that the associated likelihood based on the transformed data, $\boldsymbol{FY}$, gives the same result for any chosen contrast error matrix $\boldsymbol{F}$ and, consequently, this is invariant to fixed effect parameters (Harville 1974). Equivalently, the REML of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ can be defined as those that maximize the

$$
\ell_R(\boldsymbol{\beta},\boldsymbol{\theta};\boldsymbol{y}) = -\frac{1}{2}\log\left(\left|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\right|\right) - \frac{1}{2}\log\left(|\boldsymbol{V}|\right) - \frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^{\mathrm{T}}\boldsymbol{V}^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})
$$

This objective function is like the natural logarithm of likelihood function given in Eq. (5.2) (log-likelihood) except for the first term. To obtain the REML solutions or the maximum a posteriory (when adopting a locally uniform prior for the parameter, as described before) of the variance components parameters, (like for the MLE) numerical methods are required. See Jennrich and Schluchter (1986) and Lindstrom and Bates (1988) for details on the Newton–Raphson and Fisher Scoring algorithms; consult the lme4 R package (Bates et al. 2015) which uses a generic nonlinear optimizer and implements a large variety of different models (MLE and REML) that arise from the LMM when different structures of the variance of random effects and errors are adopted. For a derivation of the EM algorithm to obtain the REML, see Laird and Ware (1982) under model (5.1) for longitudinal data; this same approach can be used for the genomic model previously described, where $\boldsymbol{D} = \mathrm{Diag}\left(\sigma_1^2\boldsymbol{A}_1,\ \ldots,,\sigma_K^2\,\boldsymbol{A}_K\right)$ and $\boldsymbol{R} = \sigma^2\boldsymbol{I}_n$. Consult Searle (1993) and Covarrubias-Pazaran (2016, 2018) for an implementation of this algorithm with the EM function in the sommer R package.

5.2.1.3 BLUPs

In many situations, in addition to the estimation of the fixed effects, the prediction of the random effects is also of interest. A standard method for "estimating" the random effects is the best linear unbiased predictor (BLUP; Robinson 1991), which originally was developed by Henderson (1975) in animal breeding for estimating merit in dairy cattle and is now commonly employed in many research areas (Piepho et al. 2008). If the variance components $\boldsymbol{D}$ and $\boldsymbol{R}$ are known, the best linear unbiased predictor (BLUP) of the random effects $\boldsymbol{b}$ is given by

$$\widetilde{\boldsymbol{b}}^{*} = \boldsymbol{D}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{V}^{-1}\left(\boldsymbol{y}-\boldsymbol{X}\widetilde{\boldsymbol{\beta}}\right),$$

where $\widetilde{\boldsymbol{\beta}} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{y}$ is the generalized least squared (GLS) estimator of $\boldsymbol{\beta}$. This can be obtained by maximizing with respect to $\boldsymbol{\beta}$ and $\boldsymbol{b}$, the joint density of $\boldsymbol{y}$ and $\boldsymbol{b}$, $f_{\boldsymbol{Y},\boldsymbol{b}}(\boldsymbol{y},\boldsymbol{b})$, and is the reason why Harville (1985) called these estimates of realized values $\boldsymbol{b}$ (McLean et al. 1991), or likewise by solving the mixed model equations (MME) (Henderson 1975):

$$\begin{bmatrix} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{X} & \boldsymbol{X}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{Z} \\ \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{X} & \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{Z}+\boldsymbol{D}^{-1} \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{\beta}} \\ \widetilde{\boldsymbol{b}}^{*} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{y} \\ \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{y} \end{bmatrix}$$

from which the inversion of the variance–covariance matrix of $\boldsymbol{Y}$ is avoided, which can be helpful in some situations to save considerable computational resources. Note that $\widetilde{\boldsymbol{b}}^{*}$ corresponds to the posterior mean of the random effects where the fixed effects are replaced by its GLS (Searle et al. 2006).

When the variance components are unknown, which is most often the case, they are frequently estimated using restricted maximum likelihood estimators, which replace them in the corresponding equations. Then, the approximate best linear unbiased predictor is obtained and is referred to as the estimated or empirical best linear unbiased predictor (EBLUP) (Rencher 2008).

To solve the mixed model equations, there are several software packages that can be useful, but one in particular in the genomic context is the sommer package (Covarrubias-Pazaran 2016, 2018) that internally solves the MME after the variance components are estimated. The github version of the sommer R package can be accessed at https://github.com/cran/sommer and can be installed with the following commands:

```
install.packages('devtools');
 library(devtools);
 install_github('covaruber/sommer')
```

5.3 Linear Mixed Models in Genomic Prediction

In a simple genomic prediction context where $\boldsymbol{b}$ includes the genotype effects, and the genomic relationship matrix (VanRaden 2008), that is, $\boldsymbol{G}$ is available, very often the assumed variance–covariance matrix of the random effects is $\sigma_g^2\boldsymbol{G}$ and the errors are assumed as independently and identically distributed, $\boldsymbol{R} = \sigma^2\boldsymbol{I}_n$, where n is the total number of observations. In this case, the resultant model is known as the GBLUP model and when the pedigree is used, it is referred to as PBLUP. Another kind of information between lines can also be used, such as the relationship matrices derived from hyperspectral reflectance information (Krause et al. 2019). Other extensions of this model can be developed by taking into account other factors, for example, genotype× environment interaction, as will be illustrated later in the genomic prediction context.

In this case, where only the genotypic effects are taken into account, in the linear mixed model (5.1), the fixed effects design matrix is $\boldsymbol{X} = \boldsymbol{1}_n$, where the vector of length n corresponds to the general mean $\boldsymbol{\beta} = \beta_0$, $\boldsymbol{b} = (b_1, b_2, \ldots, b_J)^{\mathrm{T}}$ contains the genotypic effects of J lines, and $\boldsymbol{Z}$ is the incidence matrix design for the random line effects ($\boldsymbol{Z}_L$):

$$\boldsymbol{Y} = \boldsymbol{1}_n\mu + \boldsymbol{Z}_L\boldsymbol{b} + \boldsymbol{\epsilon}, \tag{5.3}$$

where $\boldsymbol{b} \sim N_J\left(\boldsymbol{0}, \sigma_g^2\boldsymbol{G}\right)$ and $\boldsymbol{R} = \sigma^2\boldsymbol{I}_n$.

The basic code to implement the GBLUP model (5.3) with the sommer package is the following:

```
A = mmer(y ~ 1,random= ~ vs(GID,Gu=G) , rcov= ~ vs(units),
          data=dat_F, verbose=FALSE)
```

where **y** and **GID** are the column names that contain the response variable and genotypes in data set **dat_F**. **G** is the genomic relationship matrix for lines which is specified in the Gu argument, that in general serves to provide a known variance–covariance matrix between the levels of the random effects (GID). In the “rcov” option, the argument “units” is always used to specify the error term.

5.4 Illustrative Examples of the Univariate LMM

Example 1 To illustrate the performance of the LMM in a genomic prediction context doing the fitting process with the sommer package, we considered a wheat data set that consisted of 500 markers measured for each line as the genomic information, and with 229 observations in total that registered grain yield (tons/ha): 30 lines in four environments with one or two repetitions.

Table 5.1 Prediction performance of the GBLUP model (5.3, M1) and the model (5.3) that results from ignoring the genomic information (M10): mean squared error of prediction (MSE) and Pearson's correlation (PC), and its standard deviation for each criterion is reported for each partition

	M1		M10	
PT	MSE	PC	MSE	PC
1	0.7025	0.6917	0.6924	0.6969
2	0.4583	0.6681	0.4407	0.7044
3	0.5075	0.5251	0.4923	0.5506
4	0.4719	0.6024	0.4468	0.6328
5	0.6433	0.5479	0.6433	0.5307
6	0.3657	0.5088	0.3569	0.5255
7	0.6923	0.5081	0.6777	0.5243
8	0.3285	0.5054	0.2952	0.57
9	0.4364	0.6855	0.429	0.6962
10	0.6807	0.6374	0.6513	0.6811
Average (SD)	0.5287 (0.14)	0.5881 (0.078)	0.5126 (0.143)	0.6112 (0.078)

The prediction performance of the model given in Eq. (5.3) (**M1**) was evaluated with 10 random partitions, where each partition was made up of two subsets, one containing 80% of the data and used for training the model, and the other containing the remaining 20% of the data and used to evaluate the prediction performance of the model in terms of the mean squared error of prediction (MSE).

Furthermore, this model assumes that the errors were independently and identically distributed as $\boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \sigma^2 \boldsymbol{I}_n)$; independently of the genotypic effects, $\boldsymbol{b}$ was assumed multivariate normal with a null mean vector and a variance–covariance matrix equal to $\sigma_g^2 \boldsymbol{G}$, where the genomic relationship matrix was computed with the information of the 500 markers.

The variance components parameters, in this case σ_g^2 and σ^2, were estimated by restricted maximum likelihood estimation with the mmer function in the sommer package, using the default algorithm optimization, the Newton–Raphson method. For univariate response variables, the EM algorithm through the EM function in this R package can also be used.

The results are shown in Table 5.1, where we also present the Pearson's correlation (PC) and MSE of the same model but without taking into account the information of the genomic relationship between lines ($\boldsymbol{G}$), that is, the variance–covariance matrix for the genotypic effects is assumed to be $\mathrm{Var}(\boldsymbol{b}) = \sigma_g^2 \boldsymbol{I}_J$. This model is referred to as **M10**. From this table, we can observe that model **M10** shows a slightly better performance in terms of both MSE and PC criteria than the **M1** model: the MSE of model **M1** is 3.15% greater than the MSE of **M10**, while the PC of the **M10** is 3.94% greater than the corresponding **M1** model. The better average performance was observed with model **M10**, which did not consider genomic information, suggests that the marker information in this particular case did not provide useful information; however, in general, this is not expected when using marker information for prediction, although this could change with larger data sets (more lines and more markers) or by improving the quality of the available data.

The R code to reproduce this result is given in Appendix 4. This can be adapted easily to another CV strategy of interest where the objective, for example, can be the prediction of non-observed lines in some environments or the prediction of lines in a future year.

An extension of the GBLUP model is the G×E BLUP model that takes into account the main environmental effects, the genotypic effects, and the genotype ×environment interaction effects:

$$Y = \mathbf{1}_n\mu + \boldsymbol{X}_E\boldsymbol{\beta}_E + \boldsymbol{Z}_L\boldsymbol{b}_1 + \boldsymbol{Z}_{EL}\boldsymbol{b}_2 + \boldsymbol{\epsilon} \tag{5.4}$$

where now the fixed effects are part of the linear mixed model (5.1) that was explicitly split into the general mean part ($\mathbf{1}_n\mu$) and the environment effects term ($\boldsymbol{X}_E\boldsymbol{\beta}_E$), $\boldsymbol{X} = [\mathbf{1}_n\, \boldsymbol{X}_E]$ and $\boldsymbol{\beta} = \left(\mu, \boldsymbol{\beta}_E^{\mathrm{T}}\right)^{\mathrm{T}}$. Similarly, for the random effects, $\boldsymbol{Z} = [\boldsymbol{Z}_L\, \boldsymbol{Z}_{EL}]$ and $\boldsymbol{b} = \left[\boldsymbol{b}_1^{\mathrm{T}}, \boldsymbol{b}_2^{\mathrm{T}}\right]^{\mathrm{T}}$, where $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ were the vectors with the random genotypic effects and the vector with the random genotype×environment interaction effects, with incidence matrix $\boldsymbol{Z}_L$ and $\boldsymbol{Z}_{EL}$, respectively. For $\boldsymbol{b}_1$, the same distribution as the GBLUP model was assumed, $\boldsymbol{b}_1 \sim N_J\left(\mathbf{0}, \sigma_g^2\boldsymbol{G}\right)$, and for the second random effect, $\boldsymbol{b}_2{\sim}N_J(\mathbf{0}, \boldsymbol{\Sigma}_E \otimes \boldsymbol{G})$, where $\boldsymbol{\Sigma}_E \otimes \boldsymbol{G}$ is the relationship matrix of the genotype×environment interaction term, with $\boldsymbol{\Sigma}_E$ the genetic variance–covariance matrix between I environments; the ith element of the diagonal of $\boldsymbol{\Sigma}_E$, σ_{Ei}^2, is the genetic variance in environment i, $i = 1, \ldots, I$, and $\sigma_{Eik}\boldsymbol{G}$ is the genetic variance–covariance matrix for lines in environments i and k, where σ_{Eik} is the element (i, k) of $\boldsymbol{\Sigma}_E$.

When $\boldsymbol{\Sigma}_E$ has a non-diagonal structure, the information from the genomic relationship matrix and the correlated environments can be helpful for improving the prediction performance of the model by borrowing information between lines inside an environment and between lines across and among environments (Burgueño et al. 2012).

Example 2 To illustrate how model (5.4) can be implemented using the sommer package, the same data used in Example 1 are considered, where the same 30 genotypes are in the four environments. Besides the line indicator (GID), environment information (Env) was also available in the data set, which was needed for implementing model (5.4). The adopted structure for the variance–covariance matrix between environments is $\boldsymbol{\Sigma}_E = \sigma_{EG}^2\boldsymbol{I}_I$ and the resulting model is referred to as M2. Another explored model (M20) was obtained under the same specification, with the difference that $\boldsymbol{G}$ was set equal to the identity matrix.

Using the same validation scheme that was used in Example 1, the results for each of the 10 random partitions are shown in Table 5.2, in which, for illustrative purposes, model (5.3) plus environment as a fixed effect (M11) is also included, that is,

Table 5.2 Prediction performance of two sub-models of (5.4): model M2 in which $\mathrm{Var}(\boldsymbol{b}_1) = \sigma_g^2\boldsymbol{G}$, $\boldsymbol{b}_2 \sim N_J(\mathbf{0}, \boldsymbol{\Sigma}_E \otimes \boldsymbol{G})$ and $\boldsymbol{\Sigma}_E = \sigma_{EG}^2\boldsymbol{I}_I$ (M2); and model M20 that is the same as model M2 but the genomic information is not taken into account, that is, $\boldsymbol{G} = \boldsymbol{I}_J$

	M2		M20		M11	
PT	MSE	PC	MSE	PC	MSE	PC
1	0.5641	0.8198	0.5364	0.8576	0.5658	0.8168
2	0.4292	0.7645	0.4014	0.8393	0.4447	0.6424
3	0.3865	0.7873	0.3621	0.8497	0.4246	0.6422
4	0.45	0.6997	0.401	0.7897	0.3853	0.6992
5	0.6879	0.5307	0.6636	0.5902	0.595	0.5933
6	0.3048	0.6802	0.2819	0.7326	0.3267	0.5777
7	0.6405	0.6823	0.6369	0.7066	0.558	0.6914
8	0.4132	0.5261	0.4053	0.5562	0.2999	0.5817
9	0.3217	0.8384	0.2978	0.8869	0.3457	0.7797
10	0.5397	0.8368	0.5206	0.8572	0.5647	0.7307
Average (SD)	0.4738 (0.13)	0.7166 (0.116)	0.4507 (0.133)	0.7666 (0.117)	0.451 (0.112)	0.6755 (0.083)

M11 is referred to as model (5.3) plus environment effects (Env). The mean squared error of prediction (MSE) and Pearson's correlation (PC) for each partition are reported. SD is the standard deviation

$$\boldsymbol{Y} = \mathbf{1}_n\mu + \boldsymbol{X}_E\boldsymbol{\beta}_E + \boldsymbol{Z}_L\boldsymbol{b} + \boldsymbol{\epsilon},$$

where μ, $\boldsymbol{\beta}_E$, and $\boldsymbol{b}$ are as before (5.3), and $\boldsymbol{X}_E\boldsymbol{\beta}_E$ is the predictor term corresponding to the environment fixed effects.

From Table 5.2 we can observe yet again a moderately better performance of model M20 that does not take into account the genomic information. Model M2 also confirms the lack of usefulness of the marker information in this case, but again, this in general is expected to change for other data sets with a greater number of lines, markers, or more data quality. The MSE of model M2 is 5.11% greater than the MSE of model M20, while the PC value of this last model is 6.98% greater than the corresponding PC value obtained with the M20 model. When comparing the M20 model and M11, the MSE of this last model is just 0.075% greater than the corresponding MSE of M20, but when considering the PC value, the first model resulted in 13.98% greater than model (5.1) plus the environment effect. Indeed, because of the high variation observed across partitions (SD values of PC and MSE), there is no significant difference between models in Table 5.2.

Furthermore, in terms of the average MSE, the model with the best performance between those presented in Table 5.1 (M10) is 13.73% greater than the average MSE of the best performance model between those compared in Table 5.2 (M11), while in terms of the average Pearson's correlation, the best model in Table 5.2 (M20) is 25.42% greater than the average Pearson's correlation of the best model in Table 5.1. The worse average MSE performance of those in Table 5.1 (M1) is 17.31% greater than the best average MSE performance in Table 5.2 (M20), and the best average PC performance in Table 5.2 (M20) is 30.37% greater than the worse average PC performance in Table 5.1 (M1). Actually, the best average MSE model in Table 5.1 (M10) is 8.19 greater than the worse average MSE model in Table 5.2 (M2), while the worse average PC model in Table 5.2 (M11) is 10.51% greater than the average PC model in Table 5.1 (M10).

The R code to reproduce these results is given in Appendix 5.

Other versions of model (5.4) can be obtained by adopting other variance–covariance structures. For example, another version of model (5.4) can be obtained when environment covariates are available ($\boldsymbol{W}$) and they are used to model the $G\times E$ predictor term, specifically when the genetic variance–covariance matrix between environments, $\boldsymbol{\Sigma}_E$, is modeled by $\boldsymbol{\Sigma}_E=\sigma_{EG}^2\boldsymbol{O}$, where $\boldsymbol{O}=\frac{1}{p_w}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}$ and the similarity between environments is computed like the genomic relationship matrix ($\mathbf{G}$), using the information of p_w environment covariates (Jarquín et al. 2014; Martini et al. 2020), or where $\boldsymbol{O}$ is obtained from phenotypic correlations across environments from related historical data (Martini et al. 2020). In the sommer package, this can be implemented using the following basic R code:

```
O = diag(I) #Specified the O matrix for the I environments
dat_F$Env_GID = paste(dat_F$Env,dat_F$GID,sep='_')
GE = kronecker(O,G)
rnGWE = expand.grid(row.names(G),unique(dat_F$Env))
row.names(GE) = paste(rnGWE[,2],rnGWE[,1],sep='_')
colnames(GE) = row.names(GE)
A = mmer(y ~ Env, random= ~ vs(GID,Gu=G) + vs(Env_GID,Gu = GE),
      rcov= ~ vs(units), data=dat_F)
```

Other more complex models can be explored with the sommer package when more data information is available, such as specifying an unstructured variance–covariance matrix for $\boldsymbol{\Sigma}_E$. A simpler model is the non-correlated heterogeneous variance components (for environments) which arises by assuming a diagonal structure, $\boldsymbol{\Sigma}_E=\mathrm{Diag}\left(\sigma_1^2,\ldots,\sigma_I^2\right)$. This can be implemented by replacing the interaction term in the predictor in sommer *vs(**Env**,Gu = **G**)* by *vs(ds(**Env**), **GID**,Gu = **G**)*. Similarly, for a specific environment residual variance*, vs(units)* need to be replaced by *vs(ds(Env),units)* or *vs(at(Env),units)*. See Appendix 7 for a basic code to implement all these models and see Covarrubias-Pazaran (2018) for more variance structures that can be exploited in this model.

5.5 Multi-trait Genomic Linear Mixed-Effects Models

In some genomic applications, there are several traits of interest and all of them are measured in some lines but in other lines only subsets of those traits are measured. Although separate univariate genomic linear mixed models can be performed to analyze all measured traits, sometimes single univariate genomic models do not work well, especially in traits with low heritability. When low heritability traits have at least moderate correlation with high heritability traits, the prediction performance ability for these low heritability traits could strongly increase by using a multi-trait model (Jia and Jannink 2012; Montesinos-López et al. 2016; Budhlakoti et al. 2019).

If for each line ($j = 1, \ldots J$), n_T traits are measured, Y_{jt}, $t = 1, \ldots n_T$, the multi-trait genomic linear mixed-effects model adopts an unstructured covariance matrix for the residuals between traits and for the random genotypic effects between traits, and similar to the univariate trait models (5.3), this can be expressed as

$$\begin{bmatrix} Y_{j1} \\ Y_{j2} \\ \vdots \\ Y_{jn_T} \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_{n_T} \end{bmatrix} + \begin{bmatrix} g_{j1} \\ g_{j2} \\ \vdots \\ g_{jn_T} \end{bmatrix} + \begin{bmatrix} \epsilon_{j1} \\ \epsilon_{j2} \\ \vdots \\ \epsilon_{jn_T} \end{bmatrix}, j = 1, \ldots, J, \tag{5.5}$$

where μ_t, $t = 1, \ldots, n_T$, are the specific trait means, g_{jt}, $t = 1, \ldots, n_T$, are the specific trait genotypic effects, and ϵ_{jt}, $t = 1, \ldots, n_T$, are the random error terms corresponding to each trait. Furthermore, $\boldsymbol{b} = \left[\boldsymbol{g}_1^{\mathrm{T}}, \ldots, \boldsymbol{g}_J^{\mathrm{T}}\right]^{\mathrm{T}} \sim N(\boldsymbol{0}, \boldsymbol{G} \otimes \boldsymbol{\Sigma}_{\mathrm{T}})$, $\boldsymbol{g}_j = \left[g_{j1}, \ldots, g_{jn_T}\right]^{\mathrm{T}}$, $j = 1, \ldots, J$, and $\boldsymbol{\epsilon}_j = \left[\epsilon_{j1}, \ldots, \epsilon_{jn_T}\right]^{\mathrm{T}}$, $j = 1, \ldots, J$, are independent multivariate normal random vectors with null mean and variance $\mathbf{R}_{n_T}$, $\boldsymbol{\Sigma}_T$ is $n_T \times n_T$ matrix that represents the genetic covariance between traits, and ⊗ is the Kronecker product.

In matrix notation, it is the linear mixed model (5.1) where $\boldsymbol{Y} = \left[\boldsymbol{Y}_1^{\mathrm{T}}, \ldots, \boldsymbol{Y}_J^{\mathrm{T}}\right]^{\mathrm{T}}$, $\boldsymbol{Y}_j = \left[Y_{j1}, \ldots, Y_{jn_T}\right]^{\mathrm{T}}$, $\boldsymbol{X} = \boldsymbol{1}_J \otimes \boldsymbol{I}_{\mathrm{n_T}}$, $\boldsymbol{\beta} = \boldsymbol{\mu} = \left(\mu_1, \ldots, \mu_{n_T}\right)^{\mathrm{T}}$, $\boldsymbol{Z} = \boldsymbol{I}_{n_T J}$, $\boldsymbol{b} = \left[\boldsymbol{g}_1^{\mathrm{T}}, \ldots, \boldsymbol{g}_J^{\mathrm{T}}\right]^{\mathrm{T}}, \boldsymbol{\epsilon} = \left[\boldsymbol{\epsilon}_1^{\mathrm{T}} \ldots \boldsymbol{\epsilon}_J^{\mathrm{T}}\right]^{\mathrm{T}} \sim N(\boldsymbol{0}, \boldsymbol{I}_J \otimes \boldsymbol{R}_{\mathrm{n}_T})$, and $\boldsymbol{b} = \left[\boldsymbol{g}_1^{\mathrm{T}}, \ldots, \boldsymbol{g}_J^{\mathrm{T}}\right]^{\mathrm{T}} \sim N(\boldsymbol{0}, \boldsymbol{G} \otimes \boldsymbol{\Sigma}_{\mathrm{T}})$. Similarly, the extended model that arises by adding more fixed effects ($\boldsymbol{X}$) can be specified by adding a term $\boldsymbol{X\beta}$ to the predictor:

$$\boldsymbol{Y} = (\boldsymbol{1}_{IJ} \otimes \boldsymbol{I}_{\mathrm{n_T}})\boldsymbol{\mu} + \boldsymbol{X\beta} + \boldsymbol{Zb} + \boldsymbol{\epsilon} \tag{5.5a}$$

When $\boldsymbol{\Sigma}_T$ and $\boldsymbol{R}$ are diagonal matrices, model (5.5) is equivalent to separately fitting a univariate GBLUP model to each trait.

The R code to fit this multivariate model with the sommer package is

```
A = mmer(cbind(T1,...,TnT) ~x1+x2+...+xp ,
             random= ~ vs(GID,Gu=G), rcov= ~ vs(units), data=dat_F)
```

where **y** and **GID** are again the column names corresponding to the response variables and genotypes in data set **dat_F**, while *T1*, …, *TnT* are the column names of the matrix of response variables (**y**) in **dat_F** corresponding to the traits to be used, and similarly, *x1*, *x2*,…, *xp* are the column names of *p* covariates to be included in the fitting process (see below the R code for Example 3). The rest of the arguments are the same as those described in the R code of model 5.3.

Example 3 To illustrate the fitting of the multi-trait genomic model (5.5) (M3), we considered the same data set used in Examples 1 and 2, but with the addition of trait (y2) to be able to explore the implementation of a bivariate trait genomic model. The same CV strategy implemented in Example 1 was used. In addition to model (5.5), we also evaluated a sub-model that was obtained by considering a diagonal structure for $\boldsymbol{\Sigma}_T$, that is, $\boldsymbol{\Sigma}_T = \text{Diag}\left(\sigma^2_{T_1}, \sigma^2_{T_2}\right)$. This model will be referred to as M32.

The results are in Table 5.3. On average, the two evaluated models (M3 and M32) showed a similar performance in terms of the two criteria used, MSE and PC, for both traits, but in all partitions a slightly better performance was observed in favor of model M3. For trait T1, the simpler model (M32) gave an MSE 0.785% greater than model M3, while the more complex model (M3) gave a PC only 1.066% greater than that of model M32. The difference was less for the second trait (T2), where the average MSE of M32 was only 0.165% greater than the one corresponding to model M3, while the PC of M3 was only 0.046% greater than the PC of M32.

Furthermore, note that the difference between the univariate models presented in Tables 5.1 and 5.2 and the multivariate models of Table 5.3 is not significant (only on average the models in Table 5.2 result better than models in Table 5.1) because the large standard deviation observed across partitions in MSE and PC, which in this case indicate that the multivariate model does not help improve the prediction accuracy in the trait of interest (first trait). But as commented before, this benefit could be obtained with more related auxiliary secondary traits and larger data sets of good quality.

Table 5.3 Prediction performance of a bivariate trait model (5.5)

Models	M3				M32			
Trait	T1		T2		T1		T2	
PT	MSE	PC	MSE	PC	MSE	PC	MSE	PC
1	0.6959	0.6954	0.0978	0.984	0.6973	0.6939	0.0982	0.9835
2	0.4476	0.6851	0.1411	0.9256	0.4526	0.6759	0.1409	0.9253
3	0.494	0.5448	0.0446	0.9846	0.503	0.5315	0.0436	0.985
4	0.4636	0.6125	0.1258	0.9665	0.4674	0.6079	0.1266	0.9663
5	0.6398	0.5496	0.1588	0.9306	0.6413	0.5489	0.1596	0.93
6	0.3598	0.5192	0.0691	0.9775	0.3632	0.5134	0.0693	0.9772
7	0.6864	0.514	0.0273	0.9953	0.691	0.5094	0.0272	0.9951
8	0.3104	0.5421	0.2011	0.9492	0.3187	0.5267	0.2	0.9482
9	0.4394	0.678	0.1983	0.9468	0.4359	0.6833	0.2005	0.9452
10	0.6685	0.6575	0.0838	0.974	0.6759	0.644	0.0837	0.9738
Average (SD)	0.5205 (0.142)	0.5998 (0.074)	0.1148 (0.061)	0.9634 (0.024)	0.5246 (0.141)	0.5935 (0.076)	0.115 (0.061)	0.963 (0.024)

This model is referred to as M3 and when assuming a diagonal structure for $\boldsymbol{\Sigma}_T$, $\boldsymbol{\Sigma}_T = \text{Diag}\left(\sigma^2_{1T}, \sigma^2_{2T}\right)$, it is referred to as M32. The mean squared error of prediction (MSE) and Pearson's correlation (PC) for each trait in each partition are reported. SD is the standard deviation

The R code to obtain the results given in Table 5.3 is provided in Appendix 6. At the end of this Appendix, in the comment lines, the code is also available for a CV strategy, when we are interested in evaluating the performance of a bivariate model where only trait y2 is missing in testing data set and all the information of the other trait (y1) is available. This could be useful in real applications where the interest lies in predicting traits that are difficult or expensive to measure, with the phenotypic information of correlated traits that are easy or inexpensive to measure (Calus and Veerkamp 2011; Jiang et al. 2015). Of course, the code could be adapted for any other relevant strategy.

In a similar fashion, just as univariate genomic linear mixed model (5.4), model (5.5) can be directly extended to a model that considers the genotype ×environment interaction term. Next, we do this for the balanced case, and for this we assume that for each environment $i = 1, \ldots, I$, J lines were phenotyped for n_T traits, Y_{ijt}, $t = 1, \ldots, n_T$. In matrix notation, the extended $G\times\ E$ model (5.4) plus fixed effects ($\boldsymbol{X\beta}$) is given by

$$\boldsymbol{Y} = (\mathbf{1}_{IJ}\otimes\boldsymbol{I}_{\mathrm{n_T}})\boldsymbol{\mu} + \boldsymbol{X\beta} + \boldsymbol{Z}_L\boldsymbol{b}_1 + \boldsymbol{Z}_{EL}\boldsymbol{b}_2 + \boldsymbol{\epsilon}, \tag{5.6}$$

where $\boldsymbol{Y} = \left[\boldsymbol{Y}_1^{\mathrm{T}} \ldots \boldsymbol{Y}_I^{\mathrm{T}}\right]^{\mathrm{T}}$, $\boldsymbol{Y}_i = \left[\boldsymbol{Y}_{i1}^{\mathrm{T}}, \ldots, \boldsymbol{Y}_{iJ}^{\mathrm{T}}\right]^{\mathrm{T}}$, $\boldsymbol{Y}_{ij} = \left[\boldsymbol{Y}_{ij1}, \ldots \boldsymbol{Y}_{ijn_T}\right]^{\mathrm{T}}$, $i = 1, \ldots, I$, $j = 1, \ldots, J$, $\mathbf{1}_{IJ}$ is the vector of ones of order IJ, $\boldsymbol{I}_{n_T}$ is the identity matrix of dimension n_T, $\boldsymbol{\mu} = \left(\mu_1, \ldots, \mu_{n_T}\right)^{\mathrm{T}}$ is the vector with the general specific trait means, $\boldsymbol{Z}_L = \mathbf{1}_I\otimes\boldsymbol{I}_{\mathrm{n_TJ}}$ and $\boldsymbol{Z}_{EL} = \boldsymbol{I}_{IJn_T}$ are the incidence matrices of genotype random effects ($\boldsymbol{b}_1$) and the incidence matrices of the genotype×environment interactions random effects ($\boldsymbol{b}_2$), respectively, with $\boldsymbol{b}_1 = \left[\boldsymbol{g}_1^{\mathrm{T}}, \ldots, \boldsymbol{g}_J^{\mathrm{T}}\right]^{\mathrm{T}}$ and $\boldsymbol{b_2} = \left[\boldsymbol{g}_{21}^{\mathrm{T}}, \ldots, \boldsymbol{g}_{2I}^{\mathrm{T}}\right]^{\mathrm{T}}$, $\boldsymbol{g}_j = \left[g_{j1}, \ldots, g_{jn_T}\right]^{\mathrm{T}}$, $\boldsymbol{g}_{2i} = \left[\boldsymbol{g}_{2i1}^{\mathrm{T}}, \ldots, \boldsymbol{g}_{2iJ}^{\mathrm{T}}\right]^{\mathrm{T}}$, and $\boldsymbol{g}_{2ij} = \left[\boldsymbol{g}_{2ij1}, \ldots \boldsymbol{g}_{2ijn_T}\right]^{\mathrm{T}}$, $i = 1, \ldots, I$, $j = 1, \ldots, J$. In addition, it is assumed that $\boldsymbol{\epsilon} = \left[\boldsymbol{\epsilon}_1^{\mathrm{T}} \ldots \boldsymbol{\epsilon}_I^{\mathrm{T}}\right]^{\mathrm{T}} \sim N(\mathbf{0}, \boldsymbol{I}_{IJ}\otimes\boldsymbol{R}_{\mathrm{n}_T})$, $\boldsymbol{b}_1 \sim N(\mathbf{0}, \boldsymbol{G} \otimes \boldsymbol{\Sigma}_T)$, and $\boldsymbol{b}_2 \sim N(\mathbf{0}, \boldsymbol{\Sigma}_E \otimes \boldsymbol{G} \otimes \boldsymbol{\Sigma}_{2T})$.

This shows that when $\boldsymbol{\Sigma}_T$, $\boldsymbol{\Sigma}_{2T}$, $\boldsymbol{\Sigma}_E$, and $\boldsymbol{R}$ are diagonal matrices, model (5.6) is equivalent to separately fitting a univariate GBLUP model for each trait.

Example 4 To illustrate the fitting and evaluation process of model (5.6), we considered a data set that contains the information of two traits, for which 150 lines were phenotyped each in two environments, and given a total of 300 bivariate phenotypic data points. Also, a genomic relationship matrix for the lines is available that was computed with marker information.

The first explored model is referred to as M4 and assumes an unstructured variance–covariance matrix for all the components in model (5.6), except for the assumption that $\boldsymbol{\Sigma}_E = \boldsymbol{I}_I$, i.e., the model assumes the same variance–covariance among environments. In addition to this model (M4), three sub-models were also explored: M42, which considers a diagonal structure for the genetic variance–covariance between traits, $\boldsymbol{\Sigma}_{2T} = \mathrm{Diag}\left(\sigma_{1T_1}^2, \sigma_{1T_2}^2\right)$, model M43 in which $\boldsymbol{\Sigma}_{1T} = \mathrm{Diag}\left(\sigma_{1T_1}^2, \sigma_{1T_2}^2\right)$ and $\boldsymbol{\Sigma}_{2T} = \mathrm{Diag}\left(\sigma_{2T_1}^2, \sigma_{2T_2}^2\right)$, and model M44 which is the same as

M43 but with an assumed diagonal variance–covariance matrix structure for the error, $\boldsymbol{R} = \text{Diag}\left(\sigma_{e_1}^2, \sigma_{e_2}^2\right)$.

The results are shown in Table 5.4, from which we can observe that for trait 1 (GY), the best performance under both criteria (MSE and PC) was obtained with the more complex model: M4. For this trait (GY), the MSE of models M42, M43, and M44 were 7.53%, 8.21%, and 8.17%, respectively, greater than the MSE of

Table 5.4 Prediction performance of some sub-models of model (5.6)

Model	M4				M42			
Trait	T1 (GY)		T2 (Testwt)		T1 (GY)		T2 (Testwt)	
PT	MSE	PC	MSE	PC	MSE	PC	MSE	PC
1	0.1895	0.5962	1.0291	0.6416	0.1717	0.5691	0.9357	0.6942
2	0.2414	0.5091	0.8139	0.6466	0.2444	0.5127	0.7199	0.6965
3	0.2638	0.8089	1.2914	0.5875	0.2324	0.5661	1.0839	0.7705
4	0.475	0.1266	0.9975	0.7118	0.4856	0.101	1.0271	0.6846
5	0.2728	0.2237	0.8053	0.6897	0.272	0.2402	0.8	0.683
6	0.2919	0.7391	1.031	0.5288	0.29	0.373	0.6211	0.7809
7	0.2413	0.4145	1.0778	0.7836	0.3155	0.2617	1.1276	0.7808
8	0.1696	0.7954	1.0456	0.538	0.1785	0.4238	1.1182	0.726
9	0.2412	0.8813	1.0937	0.4014	0.3293	0.3998	1.9201	0.6492
10	0.2078	0.9006	1.4155	0.3746	0.2704	0.6108	1.7571	0.6505
Average (SD)	0.2594 (0.085)	0.5995 (0.275)	1.0601 (0.186)	0.5904 (0.132)	0.279 (0.089)	0.4058 (0.166)	1.1111 (0.422)	0.7116 (0.051)
Model	M43				M44			
Trait	T1 (GY)		T2 (Testwt)		T1 (GY)		T2 (Testwt)	
PT	MSE	PC	MSE	PC	MSE	PC	MSE	PC
1	0.1717	0.5602	0.9354	0.7015	0.1716	0.5438	0.9623	0.7053
2	0.25	0.4885	0.7258	0.6987	0.2515	0.4383	0.7536	0.6907
3	0.2344	0.5482	1.1496	0.7698	0.2324	0.5182	1.2674	0.7631
4	0.4956	0.0226	1.0861	0.6277	0.4852	−0.055	1.1752	0.5615
5	0.274	0.2154	0.806	0.6778	0.2738	0.1416	0.8357	0.66
6	0.2895	0.3758	0.6211	0.7803	0.2909	0.3468	0.6096	0.7864
7	0.2925	0.319	1.0604	0.7936	0.3001	0.2746	1.0997	0.7856
8	0.1782	0.4247	1.1177	0.7261	0.1828	0.3739	1.1289	0.7139
9	0.324	0.4071	1.9091	0.6488	0.3306	0.3438	1.9212	0.6499
10	0.2974	0.5764	1.8732	0.6506	0.2874	0.5483	1.9158	0.6434
Average (SD)	0.2807 (0.091)	0.3938 (0.173)	1.1284 (0.439)	0.7075 (0.059)	0.2806 (0.088)	0.3474 (0.191)	1.1669 (0.445)	0.696 (0.071)

Model (5.6) with $\boldsymbol{\Sigma}_E = \boldsymbol{I}_I$ is referred to as M4, and if additionally, $\boldsymbol{\Sigma}_{2T} = \text{Diag}\left(\sigma_{1T_1}^2, \sigma_{1T_2}^2\right)$, the model is referred to as M42. Model M42 but with $\boldsymbol{\Sigma}_{1T} = \text{Diag}\left(\sigma_{1T_1}^2, \sigma_{1T_2}^2\right)$ is referred to as M43, and model M44 is the same as M43 but with $\boldsymbol{R} = \text{Diag}\left(\sigma_{e_1}^2, \sigma_{e_2}^2\right)$. The mean squared error of prediction (MSE) and Pearson's correlation (PC) for each trait in each partition are reported. SD is the standard deviation model M4. For the same trait (GY), the PC of model M4 was 47.73%, 52.24%, and 72.57%, greater than the PC of models M42, M43, and M44, respectively.

For the second trait (Testwt), model M4 also showed the best performance, but only under the MSE criteria: the MSE of models M42, M43, and M44 were 4.81%, 6.44%, and 10.08%, respectively, greater than the MSE corresponding to model M4, also suggesting an increasing degradation pattern in the MSE performance as the model became simpler with fewer parameters to estimate in relation to M4. In terms of PC, the best performance was achieved with model M42, which gave 20.54%, 0.583%, and 2.247% greater performances than models M4, M43, and M44, respectively.

Appendix 7 shows the R code used to reproduce the results in Table 5.4 with the sommer package. At the end of this code, we also included the basic code to explore other variance–covariance structures. Specifically, this is the code used to explore the model with heterogeneous genetic variance–covariance matrix, $\boldsymbol{\Sigma}_{2T}$, across environments, that is, $\boldsymbol{g}_{2i} \sim N(\boldsymbol{0}, \boldsymbol{G} \otimes \boldsymbol{\Sigma}_{2iT})$, $i = 1, \ldots, I$, which are assumed independent across environments.

5.6 Final Comments

The multi-trait linear model proposed by Henderson and Quaas (1976) in animal breeding can bring benefits in comparison to single-trait modeling for the improvement of prediction accuracy when incorporating correlated traits, as well as for obtaining an optimal and simplified total merit selection index (Okeke et al. 2017).

When the goal is to predict difficult or expensive traits that are correlated with inexpensive secondary traits, the use of multi-trait models could be helpful in developing better genomic selection strategies. Similarly, improvement of the accuracy of prediction for low-heritability key traits can follow from the use of high-heritability secondary traits (Jia and Jannink 2012; Muranty et al. 2015). Furthermore, this can be combined with the information of traits obtained using the speed breeding methodology to shorten the breeding cycles and accelerate breeding programs (Ghosh et al. 2018; Watson et al. 2019).

While the advantage of the multi-trait model is clearly documented, larger data sets and more computing resources are required, as there are additional parameters that need to be estimated (genetic and error covariances), which may affect the accuracy of genomic prediction. Additionally, convergence problems often arise when implementing complex mixed linear models and especially when small data sets are used.

Although the application of multi-trait models can be easily adapted with regard to genetic correlation, heritability, training population composition, and size of data sets, some other factors need to be carefully considered when using these methods for the improvement of genomic accuracy predictions (Lorenz and Smith 2015; Covarrubias-Pazaran et al. 2018). For example, biased and suboptimal choices between univariate and multi-trait models can result from using auxiliary traits that are measured on individuals to be tested, but appropriate cross-validations strategies could be helpful in determining the usefulness of combining the multi-trait information with multi-trait models (Runcie and Cheng 2019).

Appendix 1

$$\begin{aligned}
f_{\boldsymbol{b}|\boldsymbol{Y}}(\boldsymbol{b}|\boldsymbol{y}) &\propto f_{\boldsymbol{Y}|\boldsymbol{b}}(\boldsymbol{y}|\boldsymbol{b})f_{\boldsymbol{b}}(\boldsymbol{b}) \\
&\propto \frac{1}{(2\pi\sigma^2)^{-\frac{n}{2}}}\exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X\beta}-\boldsymbol{Zb})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X\beta}-\boldsymbol{Zb})-\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{D}^{-1}\boldsymbol{b}\right] \\
&\propto \exp\left[-\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\left(\boldsymbol{D}^{-1}+\sigma^{-2}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z}\right)\boldsymbol{b}-\sigma^{-2}(\boldsymbol{y}-\boldsymbol{X\beta})^{\mathrm{T}}\boldsymbol{Zb}\right] \\
&\propto \exp\left[-\frac{1}{2}\left(\boldsymbol{b}-\widetilde{\boldsymbol{b}}\right)^{\mathrm{T}}\widetilde{\boldsymbol{D}}^{-1}\left(\boldsymbol{b}-\widetilde{\boldsymbol{b}}\right)\right]
\end{aligned}$$

$\widetilde{\boldsymbol{D}}=\left(\boldsymbol{D}^{-1}+\sigma^{-2}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z}\right)^{-1}$ and $\widetilde{\boldsymbol{b}}=\sigma^{-2}\widetilde{\boldsymbol{D}}\boldsymbol{Z}^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X\beta})$. So $\boldsymbol{b} \mid \boldsymbol{Y}=\boldsymbol{y} \sim N_q\left(\widetilde{\boldsymbol{b}},\widetilde{\boldsymbol{D}}\right)$.

Appendix 2

Because $(\boldsymbol{ABA}^{\mathrm{T}}+\boldsymbol{C})^{-1}=\boldsymbol{C}^{-1}-\boldsymbol{C}^{-1}\boldsymbol{A}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{A}+\boldsymbol{B}^{-1})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{C}^{-1}$ (Johnson and Wichern (2002)), then

$$\widetilde{\boldsymbol{D}}=\left(\boldsymbol{D}+\sigma^{-2}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z}\right)^{-1}=\boldsymbol{D}-\boldsymbol{DZ}^{\mathrm{T}}\left(\boldsymbol{ZDZ}^{\mathrm{T}}+\sigma^2\boldsymbol{I}_n\right)^{-1}\boldsymbol{ZD}$$

and thus

$$\begin{aligned}
\mathrm{tr}\left(\widetilde{\boldsymbol{D}}^{(t)}\boldsymbol{D}^{-1}\right) &= \mathrm{tr}\left[\left(\boldsymbol{D}^{(t)}-\boldsymbol{D}^{(t)}\boldsymbol{Z}^{\mathrm{T}}\left(\boldsymbol{ZD}^{(t)}\boldsymbol{Z}^{\mathrm{T}}+\sigma^{2(t)}\boldsymbol{I}_n\right)^{-1}\boldsymbol{ZD}^{(t)}\right)\boldsymbol{D}^{-1}\right] \\
&= \sum_{k=1}^{K}\frac{\sigma_k^{2(t)}}{\sigma_k^2}\left[q_k-\sigma_k^{2(t)}\mathrm{tr}\left(\boldsymbol{A}_k\boldsymbol{Z}_k^T\boldsymbol{V}^{-(t)}\boldsymbol{Z}_k\right)\right],
\end{aligned}$$

where $\boldsymbol{V}^{-(t)}=(\boldsymbol{ZD}^{(t)}\boldsymbol{Z}^{\mathrm{T}}+\sigma^{2(t)}\boldsymbol{I}_n)^{-1}$.

Appendix 3

Because

$$\exp\left[-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X\beta})^{\mathrm{T}}\boldsymbol{V}^{-1}(\boldsymbol{y}-\boldsymbol{X\beta})\right]=\exp\left\{-\frac{1}{2}\left[\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X\beta}-2\boldsymbol{y}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X\beta}+\boldsymbol{y}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{y}\right]\right\}$$
$$=\exp\left\{-\frac{1}{2}\left(\boldsymbol{\beta}-\widetilde{\boldsymbol{\beta}}\right)^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\left(\boldsymbol{\beta}-\widetilde{\boldsymbol{\beta}}\right)+\frac{1}{2}\widetilde{\boldsymbol{\beta}}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\widetilde{\boldsymbol{\beta}}-\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{y}\right\},$$

where $\widetilde{\boldsymbol{\beta}}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{y}$,

$$\begin{aligned}
f(\boldsymbol{\theta}|\boldsymbol{y}) &\propto \int \frac{|\boldsymbol{V}|^{-\frac{1}{2}}}{(2\pi)^{\frac{n}{2}}}\exp\left[-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X\beta})^{\mathrm{T}}\boldsymbol{V}^{-1}(\boldsymbol{y}-\boldsymbol{X\beta})\right]d\boldsymbol{\beta}\\
&\propto |\boldsymbol{V}|^{-\frac{1}{2}}\left|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\right|^{-1/2}\exp\left\{-\frac{1}{2}\left[\boldsymbol{y}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{y}-\widetilde{\boldsymbol{\beta}}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\widetilde{\boldsymbol{\beta}}\right]\right\}\\
&\propto |\boldsymbol{V}|^{-\frac{1}{2}}\left|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\right|^{1/2}\exp\left\{-\frac{1}{2}\left[\boldsymbol{y}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{y}-2\widetilde{\boldsymbol{\beta}}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\widetilde{\boldsymbol{\beta}}+\widetilde{\boldsymbol{\beta}}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\widetilde{\boldsymbol{\beta}}\right]\right\}\\
&\propto |\boldsymbol{V}|^{-\frac{1}{2}}\left|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{X}\right|^{-1/2}\exp\left\{-\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{X}\widetilde{\boldsymbol{\beta}}\right)^{\mathrm{T}}\boldsymbol{V}^{-1}\left(\boldsymbol{y}-\boldsymbol{X}\widetilde{\boldsymbol{\beta}}\right)\right\}
\end{aligned}$$

Appendix 4

R code for Example 1:

```
rm(list=ls())
library(sommer)
load('dat_ls_E1.RData',verbose=TRUE)
#Phenotypic data
dat_F = dat_ls$dat_F
head(dat_F)
#Marker data
dat_M  = dat_ls$dat_M
dim(dat_M)
dat_F =  transform(dat_F, GID = as.character(GID))
head(dat_F,5)
#Matrix design of markers
Pos =  match(dat_F$GID,row.names(dat_M))
XM = dat_M[Pos,]
XM =  scale(XM)
#Genomic relationship matrix derived from markers
dat_M =  scale(dat_M)
G =  tcrossprod(dat_M)/dim(dat_M)[2]
```

```
dat_F$GID = factor(dat_F$GID,levels=row.names(G))
dat_F = dat_F[order(dat_F$Env,dat_F$GID),]

#10 random partitions
K = 10
n = dim(dat_F)[1]
set.seed(1)
PT = replicate(K,sample(n,0.20*n))

#Example 1
#GBLUP model
Tab = data.frame(PT = 1:K,MSEP = NA)
set.seed(1)
for(k in 1:K)
{
 Pos_tst = PT[,k]
 dat_F$y_NA = dat_F$y
 dat_F$y_NA[Pos_tst] = NA
 #M1
 A = mmer(y_NA ~ 1,na.method.Y='include', random= ~ vs(GID,Gu=G),
          rcov= ~ vs(units), data=dat_F,verbose=FALSE)
 #BLUPs
 #bv = A$U$`u:GID`$y_NA
 yp = fitted(A)$dataWithFitted$y_NA.fitted
 #Prediction of testing
 yp_ts = yp[Pos_tst]
 #MSEP and Cor
 Tab$MSEP[k] = mean((dat_F$y[Pos_tst]-yp_ts)^2)
 Tab$Cor[k] = cor(dat_F$y[Pos_tst],yp_ts)

 #M10
 A2 = mmer(y_NA ~ 1,na.method.Y='include', random= ~ vs(GID),
           rcov= ~ vs(units), data=dat_F,verbose=FALSE)
 yp2 = fitted(A2)$dataWithFitted$y_NA.fitted
 #Prediction of testing
 yp2_ts = yp2[Pos_tst]
 #MSEP
 Tab$MSEP10[k] = mean((dat_F$y[Pos_tst]-yp2_ts)^2)
 Tab$Cor10[k] = cor(dat_F$y[Pos_tst],yp2_ts)

}
```

Appendix 5

```
#Example 2
rm(list=ls())
library(sommer)
load('dat_ls_E1.RData',verbose=TRUE)
```

```
#Phenotypic data
dat_F = dat_ls$dat_F
head(dat_F)
#Marker data
dat_M = dat_ls$dat_M
dim(dat_M)
dat_F = transform(dat_F, GID = as.character(GID))
head(dat_F,5)
#Matrix design of markers
Pos = match(dat_F$GID,row.names(dat_M))
XM = dat_M[Pos,]
XM = scale(XM)
#Genomic relationship matrix derived from markers
dat_M = scale(dat_M)
G = tcrossprod(dat_M)/dim(dat_M)[2]

dat_F$GID = factor(dat_F$GID,levels=row.names(G))
dat_F = dat_F[order(dat_F$Env,dat_F$GID),]

#10 random partitions
K = 10
n = dim(dat_F)[1]
set.seed(1)
PT = replicate(K,sample(n,0.20*n))

#Model (5.4)
#Y = mu + Env + GID + GID:Env
Tab = data.frame(PT = 1:K,MSEP = NA)
set.seed(1)
dat_F$Env_GID = paste(dat_F$Env,dat_F$GID,sep='_')
GE = kronecker(diag(length(unique(dat_F$Env))),G)
rnGWE = expand.grid(row.names(G),unique(dat_F$Env))
row.names(GE) = paste(rnGWE[,2],rnGWE[,1],sep='_')
colnames(GE) = row.names(GE)
for(k in 1:K)
{
 Pos_tst = PT[,k]
 dat_F$y_NA = dat_F$y
 dat_F$y_NA[Pos_tst] = NA
 #M2
 A = mmer(y_NA ~ Env,na.method.Y='include',
       random= ~ vs(GID,Gu=G)+ vs(Env_GID,Gu = GE),
       rcov= ~ vs(units),
       data=dat_F,verbose=FALSE)
 yp = fitted(A)$dataWithFitted$y_NA.fitted
 #Prediction of testing
 yp_ts = yp[Pos_tst]
 #MSEP
 Tab$MSEP[k] = mean((dat_F$y[Pos_tst]-yp_ts)^2)
 Tab$Cor[k] = cor(dat_F$y[Pos_tst],yp_ts)

 #M20
 A20 = mmer(y_NA ~ Env,na.method.Y='include',
      random= ~ vs(GID)+vs(Env_GID),
      rcov= ~ vs(units),
      data=dat_F,verbose=FALSE)
 yp20 = fitted(A20)$dataWithFitted$y_NA.fitted
 #Prediction of testing
 yp20_ts = yp20[Pos_tst]
 Tab$MSEP20[k] = mean((dat_F$y[Pos_tst]-yp20_ts)^2)
 Tab$Cor20[k] = cor(dat_F$y[Pos_tst],yp20_ts)

 #M11
 A = mmer(y_NA ~ Env,na.method.Y='include',
       random= ~ vs(GID,Gu=G),
       rcov= ~ vs(units),
       data=dat_F,verbose=FALSE)
 yp = fitted(A)$dataWithFitted$y_NA.fitted
 #Prediction of testing
 yp_ts = yp[Pos_tst]
 Tab$MSEP11[k] = mean((dat_F$y[Pos_tst]-yp_ts)^2)
 Tab$Cor11[k] = cor(dat_F$y[Pos_tst],yp_ts)
 #M10a: Model1 (5.4) plus environment effects but with G = I_J
 A = mmer(y_NA ~ Env,na.method.Y='include',
      random= ~ vs(GID),
      rcov= ~ vs(units),
      data=dat_F,verbose=FALSE)
 yp = fitted(A)$dataWithFitted$y_NA.fitted
 #Prediction of testing
 yp_ts = yp[Pos_tst]
 Tab$MSEP10a[k] = mean((dat_F$y[Pos_tst]-yp_ts)^2)
 Tab$Cor10a[k] = cor(dat_F$y[Pos_tst],yp_ts)

}

#Basic code to implement model (5.4) with an unstructured form for
Sigma_E
A = mmer(y~ Env, na.method.Y='include',
         random= ~ vs(GID,Gu=G)+vs(us(Env),GID,Gu=G),
         rcov= ~ vs(units), data=dat_F)
#Basic code to implement model (5.4) with heterogeneous environment
variances
A = mmer(y ~ Env, ,
         random= ~ vs(GID,Gu=G)+
                   vs(ds(Env),GID,Gu=G),#or vs(at(Env),GID,Gu=G)
         rcov= ~ vs(units), data=dat_F)
#Y = mu + Env + GID + GID:ENV
#Y = mu + Env + GID + GID:ENV
#Basic code to implement model (5.4) with heterogeneous environment and
residuals variances
A = mmer(y ~ Env,
     random= ~ vs(GID,Gu=G) + vs(ds(Env),GID,Gu=G),
     rcov= ~ vs(ds(Env),units),
     data=dat_F)
```

Appendix 6

```
rm(list=ls())
library(sommer)
load('dat_ls_E1.RData',verbose=TRUE)
#Phenotypic data
dat_F = dat_ls$dat_F
head(dat_F)
#Marker data
dat_M  = dat_ls$dat_M
dim(dat_M)
dat_F =  transform(dat_F, GID = as.character(GID))
head(dat_F,5)
#Matrix design of markers
Pos =  match(dat_F$GID,row.names(dat_M))
XM = dat_M[Pos,]
XM =  scale(XM)
dim(XM)
n =  dim(dat_F)[1]
#Genomic relationship matrix derived from markers
dat_M =  scale(dat_M)
G =  tcrossprod(dat_M)/dim(dat_M)[2]

dat_F$GID = factor(dat_F$GID,levels=row.names(G))
dat_F =  dat_F[order(dat_F$Env,dat_F$GID),]

#10 random partitions
K =  10
set.seed(1)
PT = replicate(K,sample(n,0.20*n))



#Example 3
Tab = data.frame(PT = 1:K)
set.seed(1)
for(k in 1:K)
{
 #M3
 Pos_tst = PT[,k]
 dat_F$y_NA = dat_F$y
 dat_F$y_NA[Pos_tst] = NA
 dat_F$y2_NA = dat_F$y2
 dat_F$y2_NA[Pos_tst] = NA
 A =  mmer(cbind(y_NA,y2_NA) ~ 1,na.method.Y='include',
       random= ~ vs(GID,Gu=G) ,
       rcov= ~ vs(units),
       data=dat_F,verbose=FALSE)
 #BLUPs
 b_ls = A$U$`u:GID`
 b_T1 = b_ls$y_NA# Trait 1
 b_T2 = b_ls$y2_NA# Trait 2
 b_mat = cbind(b_T1,b_T2)
 Pos = match(dat_F$GID,names(b_ls$y_NA))
 #Y "fitted"
 yp = A$fitted + b_mat[Pos,]
 #Prediction of testing for both traits
 yp_ts = yp[Pos_tst,1]# Trait 1
 y2p_ts = yp[Pos_tst,2]# Trait 2
 #MSEP and Cor
 #Trait 1
 Tab$MSEP_T1[k] = mean((dat_F$y[Pos_tst]-yp_ts)^2)
 Tab$Cor_T1[k] = cor(dat_F$y[Pos_tst],yp_ts)
 #Trait 2
 Tab$MSEP_T2[k] = mean((dat_F$y2[Pos_tst]-y2p_ts)^2)
 Tab$Cor_T2[k] = cor(dat_F$y2[Pos_tst],y2p_ts)
 #M32: Sigma_T diagonal
 A = mmer(cbind(y_NA,y2_NA) ~ 1,na.method.Y='include',
       random= ~ vs(GID,Gu=G,Gtc=diag(2)),
       rcov= ~ vs(units),
       data=dat_F,verbose=FALSE)
 #BLUPs
 b_ls = A$U$`u:GID`
 b_T1 = b_ls$y_NA# Trait 1
 b_T2 = b_ls$y2_NA# Trait 2
 b_mat = cbind(b_T1,b_T2)
 Pos = match(dat_F$GID,names(b_ls$y_NA))
 #Y "fitted"
 yp = A$fitted + b_mat[Pos,]
 #Prediction of testing
 yp_ts = yp[Pos_tst,1]# Trait 1
 y2p_ts = yp[Pos_tst,2]# Trait 2
 #MSEP and Cor
 #Trait 1
 Tab$MSEP_T1_32[k] = mean((dat_F$y[Pos_tst]-yp_ts)^2)
 Tab$Cor_T1_32[k] = cor(dat_F$y[Pos_tst],yp_ts)
#Trait 2
 Tab$MSEP_T2_32[k] = mean((dat_F$y2[Pos_tst]-y2p_ts)^2)
 Tab$Cor_T2_32[k] = cor(dat_F$y2[Pos_tst],y2p_ts)

 # #M3
 # #dat_F$y2_NA = dat_F$y2
 # dat_F$y_NA = dat_F$y# Trait T1 is not NA
 # A = mmer(cbind(y_NA,y2_NA) ~ 1,na.method.Y='include',
 #       random= ~ vs(GID,Gu=G),
 #       rcov= ~ vs(units),#tolparinv = 1e-2,
 #       data=dat_F,verbose=FALSE)
 # #BLUPs
 # b_ls = A$U$`u:GID`
 # b_T1 = b_ls$y_NA# Trait 1
 # b_T2 = b_ls$y2_NA# Trait 2
 # b_mat = cbind(b_T1,b_T2)
 # Pos = match(dat_F$GID,names(b_ls$y_NA))
 # #Y "fitted"
 # yp = A$fitted + b_mat[Pos,]
 # #Prediction of testing
 # yp_ts = yp[Pos_tst,1]# Trait 1
 # y2p_ts = yp[Pos_tst,2]# Trait 2
 # #MSEP
 # Tab$MSEP_T1_31[k] = mean((dat_F$y[Pos_tst]-yp_ts)^2)
 # Tab$Cor_T1_31[k] = cor(dat_F$y[Pos_tst],yp_ts)
 # Tab$MSEP_T2_31[k] = mean((dat_F$y2[Pos_tst]-y2p_ts)^2)
 # Tab$Cor_T2_31[k] = cor(dat_F$y2[Pos_tst],y2p_ts)
}
```

Appendix 7

```
#Example 4
rm(list=ls())
library(sommer)
load('dat_ls_E4.RData',verbose=TRUE)
#Phenotypic data
dat_F = dat_ls$dat_F
dat_F = transform(dat_F, GID = as.character(GID))
#Genomic relationship matrix derived from markers
G = dat_ls$G
dat_F$GID = factor(dat_F$GID,levels=row.names(G))
dat_F = dat_F[order(dat_F$Env,dat_F$GID),]
#Fitting model M4 with data set
#A = mmer(cbind(GY,TESTWT) ~ Env,na.method.Y='include',
#      random= ~ vs(GID,Gu=G) + vs(Env_GID,Gu=GE) ,
#      rcov= ~ vs(units),
#      data=dat_F,verbose=FALSE)
#Example 4
#10 random partitions
n = dim(dat_F)[1] ;K = 10
set.seed(1)
PT = replicate(K,sample(n,0.20*n))
#Model 5.6
#Y = mu + Env + GID + GID:Env + e
Tab = data.frame(PT = 1:K)
set.seed(1)
dat_F$Env_GID = paste(dat_F$Env,dat_F$GID,sep='_')
GE = kronecker(diag(length(unique(dat_F$Env))),G)
rnGWE = expand.grid(row.names(G),unique(dat_F$Env))
row.names(GE) = paste(rnGWE[,2],rnGWE[,1],sep='_')
colnames(GE) = row.names(GE)
for(k in 1:K)
{
 Pos_tst = PT[,k]
 #M4
 Pos_tst = PT[,k]
 dat_F$GY_NA = dat_F$GY
 dat_F$GY_NA[Pos_tst] = NA
 dat_F$TESTWT_NA = dat_F$TESTWT
 dat_F$TESTWT_NA[Pos_tst] = NA
 A = mmer(cbind(GY_NA,TESTWT_NA ) ~ Env,na.method.Y='include',
       random= ~ vs(GID,Gu=G) + vs(Env_GID,Gu=GE) ,
       rcov= ~ vs(units),tolparinv = 1,
       data=dat_F,verbose=FALSE)
 #BLUPs
 b_ls = A$U$`u:GID`
 b_T1 = b_ls$GY_NA# Trait 1
 b_T2 = b_ls$TESTWT_NA# Trait 2
 b_mat = cbind(b_T1,b_T2)
 Pos = match(dat_F$GID,names(b_ls$TESTWT))
 #Y "fitted"
 yp = data.frame(A$fitted + b_mat[Pos,] )
 colnames(yp) = names(b_ls)
 plot(dat_F$GY,yp$GY_NA);abline(a=0,b=1)
 #Prediction of testing of both traits
 yp_tst = yp[Pos_tst,]
 #plot(dat_F$y2,yp[,2]); abline(a=0,b=1)
 #MSEP and Cor
 #Trait 1
 Tab$MSEPT1[k] = mean((dat_F$GY[Pos_tst]-yp_tst$GY_NA)^2)
 Tab$CorT1[k] = cor(dat_F$GY[Pos_tst],yp_tst$TESTWT_NA)
 #Trait 2
 Tab$MSEPT2[k] = mean((dat_F$TESTWT[Pos_tst]-yp_tst$TESTWT_NA)^2)
 Tab$CorT2[k] = cor(dat_F$TESTWT[Pos_tst],yp_tst$TESTWT_NA)

 #M42
 A42 = mmer(cbind(GY_NA,TESTWT_NA) ~ Env,na.method.Y='include',
        random= ~ vs(GID,Gu=G) + vs(Env_GID,Gu=GE,Gtc=diag(2)) ,
        rcov= ~ vs(units),
        data=dat_F,verbose=FALSE)
 #BLUPs
 b_ls = A42$U$`u:GID`
 b_T1 = b_ls$GY_NA# Trait 1
 b_T2 = b_ls$TESTWT_NA# Trait 2
 b_mat = cbind(b_T1,b_T2)
 Pos = match(dat_F$GID,names(b_ls$TESTWT))
 #Y "fitted"
 yp = data.frame(A$fitted + b_mat[Pos,] )
 colnames(yp) = names(b_ls)
 #plot(dat_F$GY,yp$GY_NA);abline(a=0,b=1)
 #Prediction of testing of both traits
 yp_tst = yp[Pos_tst,]
 #MSEP and Cor
 #Trait 1
 Tab$MSEPT1_42[k] = mean((dat_F$GY[Pos_tst]-yp_tst$GY_NA)^2)
 Tab$CorT1_42[k] = cor(dat_F$GY[Pos_tst],yp_tst$TESTWT_NA)
 #Trait 2
 Tab$MSEPT2_42[k] = mean((dat_F$TESTWT[Pos_tst]-yp_tst$TESTWT_NA)
^2)
 Tab$CorT2_42[k] = cor(dat_F$TESTWT[Pos_tst],yp_tst$TESTWT_NA)
 #M43
 A43 = mmer(cbind(GY_NA,TESTWT_NA) ~ Env,na.method.Y='include',
        random= ~ vs(GID,Gu=G,Gtc=diag(2)) +
              vs(Env_GID,Gu=GE,Gtc=diag(2)) ,
        rcov= ~ vs(units),
        data=dat_F,verbose=FALSE)
 #BLUPs
 b_ls = A43$U$`u:GID`
 b_T1 = b_ls$GY_NA# Trait 1
 b_T2 = b_ls$TESTWT_NA# Trait 2
 b_mat = cbind(b_T1,b_T2)
 Pos = match(dat_F$GID,names(b_ls$TESTWT))
 #Y "fitted"
 yp = data.frame(A$fitted + b_mat[Pos,] )
 colnames(yp) = names(b_ls)
 #plot(dat_F$GY,yp$GY_NA);abline(a=0,b=1)
 #Prediction of testing of both traits
 yp_tst = yp[Pos_tst,]
 #MSEP and Cor
 #Trait 1
 Tab$MSEPT1_43[k] = mean((dat_F$GY[Pos_tst]-yp_tst$GY_NA)^2)
 Tab$CorT1_43[k] = cor(dat_F$GY[Pos_tst],yp_tst$TESTWT_NA)
 #Trait 2
 Tab$MSEPT2_43[k] = mean((dat_F$TESTWT[Pos_tst]-yp_tst$TESTWT_NA)
^2)
 Tab$CorT2_43[k] = cor(dat_F$TESTWT[Pos_tst],yp_tst$TESTWT_NA)
  A44 = mmer(cbind(GY_NA,TESTWT_NA) ~ Env,na.method.Y='include',
        random= ~ vs(GID,Gu=G,Gtc=diag(2)) +
         vs(Env_GID,Gu=GE,Gtc=diag(2)) ,
        rcov= ~ vs(units,Gtc=diag(2)),
        data=dat_F,verbose=FALSE)
 #A44$sigma
 #BLUPs
 b_ls = A44$U$`u:GID`
 b_T1 = b_ls$GY_NA# Trait 1
 b_T2 = b_ls$TESTWT_NA# Trait 2
 b_mat = cbind(b_T1,b_T2)
 Pos = match(dat_F$GID,names(b_ls$TESTWT))
 #Y "fitted"
 yp = data.frame(A$fitted + b_mat[Pos,] )
 colnames(yp) = names(b_ls)
 #plot(dat_F$GY,yp$GY_NA);abline(a=0,b=1)
 #Prediction of testing of both traits
 yp_tst = yp[Pos_tst,]
 #MSEP and Cor
 #Trait 1
 Tab$MSEPT1_44[k] = mean((dat_F$GY[Pos_tst]-yp_tst$GY_NA)^2)
 Tab$CorT1_44[k] = cor(dat_F$GY[Pos_tst],yp_tst$TESTWT_NA)
 #Trait 2
 Tab$MSEPT2_44[k] = mean((dat_F$TESTWT[Pos_tst]-yp_tst$TESTWT_NA)
^2)
 Tab$CorT2_44[k] = cor(dat_F$TESTWT[Pos_tst],yp_tst$TESTWT_NA)
 cat('k=',k,'\n')

}
```

```
#Model 5.6 with Sigma_2T different for each Env
A4 = mmer(cbind(GY,TESTWT)~Env,
        random = ~vs(GID,Gu=G)+vs(ds(Env),GID,Gu=G),data=dat_F,
        rcov= ~ vs(units))
```

References

Araus JL, Cairns JE (2014) Field high-throughput phenotyping: the new crop breeding frontier. Trends Plant Sci 19(1):52–61

Bates D, Maechler M, Bolker B, Walker S (2015) Fitting linear mixed-effects models using lme4. J Stat Softw 67(1):1–48

Borman S (2004) The expectation maximization algorithm: a short tutorial. https://www.lri.fr/~sebag/COURS/EM_algorithm.pdf

Brown H, Prescott R (2014) Applied mixed models in medicine. John Wiley & Sons, Hoboken, NJ

Budhlakoti N, Mishra DC, Rai A, Lal SB, Chaturvedi KK, Kumar RR (2019) A comparative study of single-trait and multi-trait genomic selection. J Comput Biol 26(10):1100–1112

Burgueño J, de los Campos G, Weigel K, Crossa J (2012) Genomic prediction of breeding values when modeling genotype × environment interaction using pedigree and dense molecular markers. Crop Sci 52(2):707–719

Cabrera-Bosquet L, Crossa J, von Zitzewitz J, Serret MD, Luis Araus J (2012) High-throughput phenotyping and genomic selection: the frontiers of crop breeding converge F. J Integr Plant Biol 54(5):312–320

Calus MP, Veerkamp RF (2011) Accuracy of multi-trait genomic selection using different methods. Genet Select Evol 43(1):26. https://doi.org/10.1186/1297-9686-43-26

Cappa EP, de Lima BM, da Silva-Junior OB, Garcia CC, Mansfield SD, Grattapaglia D (2019) Improving genomic prediction of growth and wood traits in Eucalyptus using phenotypes from non-genotyped trees by single-step GBLUP. Plant Sci 284:9–15

Covarrubias-Pazaran G (2016) Genome-assisted prediction of quantitative traits using the R package sommer. PLoS One 11(6):e0156744

Covarrubias-Pazaran G (2018) Software update: moving the R package sommer to multivariate mixed models for genome-assisted prediction. https://doi.org/10.1101/354639

Covarrubias-Pazaran G, Schlautman B, Diaz-Garcia L, Grygleski E, Polashock J, Johnson-Cicalese J et al (2018) Multivariate GBLUP improves accuracy of genomic selection for yield and fruit weight in biparental populations of Vaccinium macrocarpon Ait. Front Plant Sci 9:1310

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 Plant Sci 22(11):961–975

Finch WH, Bolin JE, Kelley K (2019) Multilevel modeling using R. CRC Press, Boca Raton, FL

Ghosh S, Watson A, Gonzalez-Navarro OE, Ramirez-Gonzalez RH, Yanes L, Mendoza-Suárez M et al (2018) Speed breeding in growth chambers and glasshouses for crop breeding and model plant research. Nat Protoc 13(12):2944–2963

Goldstein H (2011) Multilevel statistical models. Wiley, Hoboken, NJ

Harville DA (1974) Bayesian inference for variance components using only error contrasts. Biometrika 61(2):383–385

Harville DA (1977) Maximum likelihood approaches to variance component estimation and to related problems. J Am Stat Assoc 72(358):320–338

Harville DA (1985) Decomposition of prediction error. J Am Stat Assoc 80(389):132–138

Henderson CR (1950) Estimation of genetic parameters. Ann Math Stat 21:309–310

Henderson CR (1975) Best linear unbiased estimation and prediction under a selection model. Biometrics 31:423–447

Henderson CR, Quaas RL (1976) Multiple trait evaluation using relatives' records. J Anim Sci 43 (6):1188–1197

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. Theor Appl Genet 127(3):595–607

Jennrich RI, Sampson PF (1976) Newton-Raphson and related algorithms for maximum likelihood variance component estimation. Technometrics 18(1):11–17

Jennrich RI, Schluchter MD (1986) Unbalanced repeated-measures models with structured covariance matrices. Biometrics 42:805–820

Jia Y, Jannink JL (2012) Multiple-trait genomic selection methods increase genetic value prediction accuracy. Genetics 192(4):1513–1522

Jiang J, Zhang Q, Ma L, Li J, Wang Z, Liu JF (2015) Joint prediction of multiple quantitative traits using a Bayesian multivariate antedependence model. Heredity 115(1):29–36

Johnson RA, Wichern DW (2002) Applied multivariate statistical analysis. Prentice Hall, Upper Saddle River, NJ

Krause MR, González-Pérez L, Crossa J, Pérez-Rodríguez P, Montesinos-López O, Singh RP et al (2019) Hyperspectral reflectance-derived relationship matrices for genomic prediction of grain yield in wheat. G3 9(4):1231–1247

Laird NM, Ware JH (1982) Random-effects models for longitudinal data. Biometrics 38:963–974

Leyland AH, Goldstein H (2001) Multilevel modelling of health statistics. Wiley, Hoboken, NJ

Lindstrom MJ, Bates DM (1988) Newton–Raphson and EM algorithms for linear mixed-effects models for repeated-measures data. J Am Stat Assoc 83(404):1014–1022

Lorenz AJ, Smith KP (2015) Adding genetically distant individuals to training populations reduces genomic prediction accuracy in barley. Crop Sci 55(6):2657–2667

Martini JW, Crossa J, Toledo FH, Cuevas J (2020) On Hadamard and Kronecker products in covariance structures for genotype × environment interaction. Plant Genome 13:e20033

McLean RA, Sanders WL, Stroup WW (1991) A unified approach to mixed linear models. Am Stat 45(1):54–64

Meeker W, Hong Y, Escobar L (2011) Degradation models and analyses. In: Encyclopedia of statistical sciences. Wiley, Hoboken, NJ. https://doi.org/10.1002/0471667196.ess7148

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

Montesinos-López OA, Montesinos-López A, Crossa J, Toledo FH, Pérez-Hernández O, Eskridge KM, Rutkoski J (2016) A genomic Bayesian multi-trait and multi-environment model. G3 6 (9):2725–2744

Muranty H, Troggio M, Sadok IB, Al Rifaï M, Auwerkerken A, Banchi E et al (2015) Accuracy and responses of genomic selection on key traits in apple breeding. Horticult Res 2(1):1–12

Okeke UG, Akdemir D, Rabbi I, Kulakow P, Jannink JL (2017) Accuracies of univariate and multivariate genomic prediction models in African cassava. Genet Sel Evol 49(1):88

Patterson HD, Thompson R (1971) Recovery of inter-block information when block sizes are unequal. Biometrika 58:545–554

Piepho HP, Möhring J, Melchinger AE, Büchse A (2008) BLUP for phenotypic selection in plant breeding and variety testing. Euphytica 161(1–2):209–228

Pinheiro JC, Bates DM (2000) Mixed-effects models in S and S-PLUS. Springer, New York

Poland J, Endelman J, Dawson J, Rutkoski J, Wu S, Manes Y et al (2012) Genomic selection in wheat breeding using genotyping-by-sequencing. Plant Genome 5(3):103–113

Raudenbush SW, Bryk AS (2002) Hierarchical linear models: applications and data analysis methods. Sage Publications, Inc, Thousand Oaks, CA

Rencher AC (2008) Linear models in statistics. Wiley, Hoboken, NJ

Robinson GK (1991) That BLUP is a good thing: the estimation of random effects. Stat Sci 6 (1):15–32

Runcie D, Cheng H (2019) Pitfalls and remedies for cross validation with multi-trait genomic prediction methods. G3 9(11):3727–3741

Searle SR (1993) Applying the EM algorithm to calculating ML and REML estimates of variance components. In: Paper invited for the 1993 American Statistical Association Meeting, San Francisco

Searle SR, Casella G, McCulloch CE (2006) Variance components. Wiley, Hoboken, NJ

Speelman D, Heylen K, Geeraerts D (eds) (2018) Mixed-effects regression models in linguistics. Springer, New York

Stroup WW (2012) Generalized linear mixed models: modern concepts, methods and applications. CRC Press, Boca Raton, FL

VanRaden PM (2008) Efficient methods to compute genomic predictions. J Dairy Sci 91:4414–4423

Wang X, Xu Y, Hu Z, Xu C (2018) Genomic selection methods for crop improvement: current status and prospects. Crop J 6(4):330–340

Watson A, Hickey LT, Christopher J, Rutkoski J, Poland J, Hayes BJ (2019) Multivariate genomic selection and potential of rapid indirect selection with speed breeding in spring wheat. Crop Sci 59(5):1945–1959

West BT, Welch KB, Galecki AT (2014) Linear mixed models: a practical guide using statistical software. CRC Press, Boca Raton, FL

Zuur A, Ieno EN, Walker N, Saveliev AA, Smith GM (2009) Mixed effects models and extensions in ecology with R. Springer Science & Business Media, New York