

SUPPLEMENTARY MATERIAL

Table 1: Summary of methods

Method	Prior	Variable Selection	Loss function	Class
wgr	User specified	User specified	L2	MCMC
BayesA	Student's t	None	L2	MCMC
BayesB	Student's t	Unconditional prior	L2	MCMC
BayesC	Gaussian	Unconditional prior	L2	MCMC
BayesCpi	Gaussian	Metropolis-Hasting	L2	MCMC
BayesDpi	Student's t	Metropolis-Hasting	L2	MCMC
BayesL	Laplace	Laplacian	L2	MCMC
BayesRR	Gaussian	None	L2	MCMC
emBA	Student's t	None	L2	EM
emBB	Student's t	Unconditional prior	L2	EM
emBC	Gaussian	Unconditional prior	L2	EM
emBL	Laplace	L1	L1	EM
emRR	Gaussian	None	L2	EM
emDE	Laplace	Laplacian	L2	Empirical Bayes
emEN	Laplace	L1	L1-L2	Empirical Bayes
emML	Gaussian	None	L2	Empirical Bayes
mrr	MV-Gaussian	None	L2	Empirical Bayes
mkr	MV-Gaussian	None	L2	Empirical Bayes
mixed	User specified	User specified	L2	Single-step

CROSS-VALIDATION STUDIES

The functions “emCV” and “mcmcCV” enable users to perform univariate cross-validation studies that allows for the identification of models that are more suitable to the genetic architecture of any given trait. The outcome of these function is the average predictive ability as the Pearson correlation between predicted and observed values or (if avg=FALSE) the predictive ability for individual cross-validation. The latter allows for plotting box plots like what was presented in figure 1.

```
emCV(y, gen, k=5, n=5, Pi=0.75, alpha=0.02, df=10, R2=0.5, avg=TRUE, llo=NULL, tbv=NULL)
mcmcCV(y, gen, k=5, n=5, it=1500, bi=500, pi=0.95, df=5, R2=0.5, avg=T, llo=NULL, tbv=NULL)
```

The target of predictions

The prediction is target is the observed phenotypes, unless “tbv” is provided, a numeric vector with the same length as “y”, which contains the true breeding values, or true signal. When “tbv is provided”, the observations left out will not be included in the training set, and the predictive ability reported will correspond to the correlation between predicted values and TBVs.

Different types of cross-validation:

- **k-fold cross-validation:** By default, the cross-validations are performed as a standard k-fold testing controlled by the arguments “k” and “n”; by splitting the data into 1/“k” fraction of the total number of observations, randomly selecting individuals and repeating the process “n” times. The seed is set as one-through-n such that the results will be reproducible for a given combination of dataset, with the same values of “k” and “n”.
- **Leave-level-out cross-validation:** The “llo” argument allows user to provide the splits, taking as input numeric vectors, character vectors, or factors. “llo” can be used for interpretable cross-validation studies such as “leave-family-out” or “leave-population-out”, as well as to replicate cross-validation where the splits were stored. In the extreme case, if “llo” is a vector where each value is unique, then users run “leave-one-out” cross-validation.

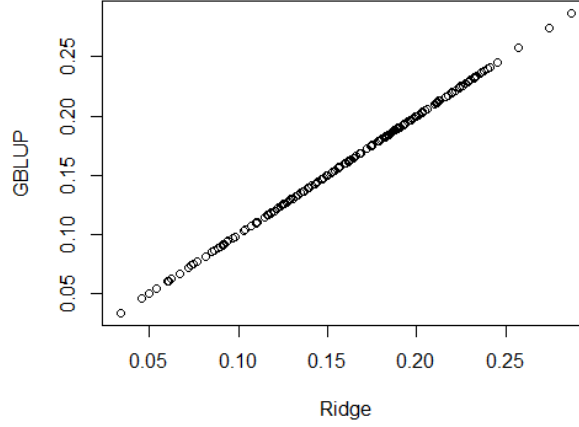
KERNEL METHODS

The bWGR package enable users to fit Kernel methods through the following functions: “emML”, “emML2”, “mkr” and “mkr2X”. Whereas functions “mkr” and “mkr2X” are explicitly built for solving kernel methods, the “emML” and “emML2” functions take the output of the spectral decomposition from the function “eigen” (Eigenvectors and Eigenvalues) to fit GBLUP and RKHS models. Below, we present the code of an example of a GBLUP that was fitted using the soybean dataset that comes with the package:

```
1 # Test dataset
2 data(tpod)
3 # Create genomic relationship matrix
4 G = GRM(gen)
5 # Eigendecomposition
6 eig = eigen(G,symmetric=TRUE)
7 # Fit kernel-based GBLUP model
8 fit_GBLUP = emML( y, eig$vectors, eig$values )
```

For comparison, we showed that fitting the GBLUP model yield identical results to its regression counterpart model:

```
9 # Compare to ridge regression BLUP
10 fit_Ridge = emML( y, gen )
11 plot(fit_Ridge$hat, fit_GBLUP$hat,ylab='GBLUP',xlab='Ridge')
```



If users have pedigree information, the function “Hmat” can be used to create a combined pedigree-genomics relationship matrix. This function builds the classical Wright’s pedigree matrix that adds one individuals at a time, but it uses Modified Rodgers’ dissimilarity when genomic information is available and estimates the inbreeding coefficient (diagonal value) as one plus the percentage of homozygous loci.

OVERVIEW OF REGRESSION METHODS

Linear model: All MCMC and EM methods follow the same Gaussian linear model:

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{X}\mathbf{b} + \mathbf{e}$$

$$\mathbf{e} \sim N(0, \mathbf{I}\sigma_e^2)$$

Prior information: What differentiates the methods is how the distribution of coefficients (b). For most Bayesian methods, variance components rely on user-specified prior degrees of freedom (v), and prior shape (S) based on the user-specified R^2 . Thus:

$$S_b = R^2 \sigma_y^2 \sum \sigma_{x_j}^2$$

$$S_e = (1 - R^2) \sigma_y^2$$

Variable selection: Models BayesB and BayesC perform variable selection using the unconditional prior proposed by Kuo and Mallick (1998), based on the ratio between the probabilities on markers included ($b \neq 0$) and marker excluded ($b = 0$).

$$p(b \neq 0) = (1 - \pi) \exp(-\mathbf{e}'_{b \neq 0} \mathbf{e}_{b \neq 0} \sigma_e^{-2})$$

$$p(b = 0) = \pi \exp(-\mathbf{e}'_{b=0} \mathbf{e}_{b=0} \sigma_e^{-2})$$

Models BayesCpi and BayesDpi perform variable selection using the Metropolis-Hasting described by Habier et al. (2011), where the probability of null effect (π) is inferred as the average of zero-effect markers from each Markov chain iteration.

Convergence: MCMC methods have as final estimates the posterior mean, that is, we average all the sampled values from burn-in (**bi**) to the total number of iterations (**it**). EM methods iteratively update the coefficients as opposed to rely on sampling. The convergence criteria for the EM methods are: changes in regression coefficients lower than 10^{-8} from an iteration to the next or 300 iterations.

Heritability: Because not all methods have a straight forward set of variance components that can be used to estimate heritability, we used a general estimator:

$$h^2 = 1 - \sigma_e^2 \sigma_y^{-2}, \quad \sigma_e^2 = \mathbf{y}'\mathbf{e}/(n - 1)$$

Algorithm: A summary of the sampling of MCMC methods and iterative updates of EM methods is provided below. Algorithms rely on iterating the intercept (μ), loop across marker effects (b_j), and solve variance components (σ_e^2 and σ_b^2 or $\sigma_{b_j}^2$).

MCMC METHODS

BayesRR

$$\begin{aligned} \mu | \text{else} &\sim N(n^{-1} \sum_i y_i - \mathbf{X}_i \mathbf{b}, n^{-1} \sigma_e^2) \\ b_j | \text{else} &\sim N\left(\frac{\mathbf{x}_j'(\mathbf{y} - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}}, \frac{\sigma_e^2}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}}\right) \\ \sigma_e^2 | \text{else} &\sim \frac{\mathbf{e}'\mathbf{e} + S_e v}{\chi^2(n + v)} \\ \sigma_b^2 | \text{else} &\sim \frac{\mathbf{b}'\mathbf{b} + S_b v}{\chi^2(p + v)} \end{aligned}$$

BayesA

$$\begin{aligned} \mu | \text{else} &\sim N(n^{-1} \sum_i y_i - \mathbf{X}_i \mathbf{b}, n^{-1} \sigma_e^2) \\ b_j | \text{else} &\sim N\left(\frac{\mathbf{x}_j'(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_{b_j}^{-2}}, \frac{\sigma_e^2}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_{b_j}^{-2}}\right) \\ \sigma_e^2 | \text{else} &\sim \frac{\mathbf{e}'\mathbf{e} + S_e v}{\chi^2(n + v)} \\ \sigma_{b_j}^2 | \text{else} &\sim \frac{b_j^2 + S_b v}{\chi^2(1 + v)} \end{aligned}$$

BayesB

$$\begin{aligned} \mu | \text{else} &\sim N(n^{-1} \sum_i y_i - \mathbf{X}_i \mathbf{b}, n^{-1} \sigma_e^2) \\ d_j | \text{else} &\sim \text{Bernoulli}\left(\frac{p(b_j \neq 0)}{p(b_j \neq 0) + p(b_j = 0)}\right) \\ \text{IF } d_j = 1: b_j | \text{else} &\sim N\left(\frac{\mathbf{x}_j'(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_{b_j}^{-2}}, \frac{\sigma_e^2}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_{b_j}^{-2}}\right) \end{aligned}$$

$$\begin{aligned}
\text{IF } d_j = 0: \quad b_j | \text{else} &\sim N\left(0, \frac{\sigma_e^2}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}}\right) \\
\sigma_e^2 | \text{else} &\sim \frac{\mathbf{e}' \mathbf{e} + S_e v}{\chi^2(n + v)} \\
\sigma_{b_j}^2 | \text{else} &\sim \frac{\mathbf{b}' \mathbf{b} + S_b v}{\chi^2(1 + v)}
\end{aligned}$$

BayesC

$$\begin{aligned}
\mu | \text{else} &\sim N(n^{-1} \sum_i y_i - 1 \mu - \mathbf{X}_1 \mathbf{b}, n^{-1} \sigma_e^2) \\
d_j | \text{else} &\sim \text{Bernoulli}\left(\frac{p(b_j \neq 0)}{p(b_j \neq 0) + p(b_j = 0)}\right) \\
\text{IF } d_j = 1: \quad b_j | \text{else} &\sim N\left(\frac{\mathbf{x}_j'(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}}, \frac{\sigma_e^2}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}}\right) \\
\text{IF } d_j = 0: \quad b_j | \text{else} &\sim N\left(0, \frac{\sigma_e^2}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}}\right) \\
\sigma_e^2 | \text{else} &\sim \frac{\mathbf{e}' \mathbf{e} + S_e v}{\chi^2(n + v)} \\
\sigma_b^2 | \text{else} &\sim \frac{\mathbf{b}' \mathbf{b} + S_b v}{\pi \chi^2(p + v)}
\end{aligned}$$

BayesCpi

$$\begin{aligned}
\mu | \text{else} &\sim N\left(n^{-1} \sum_i y_i - \mathbf{X}_1 \mathbf{b}, n^{-1} \sigma_e^2\right) \\
p_j | \text{else} &= \min\left(1, \pi \exp\left(e'_{b_j \neq 0} e_{b_j \neq 0} - e'_{b_j = 0} e_{b_j = 0}\right)\right) \\
d_j | \text{else} &\sim \text{Bernoulli}(p_j) \\
\text{IF } d_j = 1: \quad b_j | \text{else} &\sim N\left(\frac{\mathbf{x}_j'(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}}, \frac{\sigma_e^2}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}}\right) \\
\text{IF } d_j = 0: \quad b_j | \text{else} &\sim N\left(0, \frac{\sigma_e^2}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}}\right) \\
\sigma_e^2 | \text{else} &\sim \frac{\mathbf{e}' \mathbf{e} + S_e v}{\chi^2(n + v)} \\
\sigma_b^2 | \text{else} &\sim \frac{\mathbf{b}' \mathbf{b} + S_b v}{\pi \chi^2(p + v)}
\end{aligned}$$

BayesDpi

$$\begin{aligned}
\mu | \text{else} &\sim N\left(n^{-1} \sum_i y_i - \mathbf{X}_1 \mathbf{b}, n^{-1} \sigma_e^2\right) \\
p_j | \text{else} &= \min\left(1, \pi \exp\left(e'_{b_j \neq 0} e_{b_j \neq 0} - e'_{b_j = 0} e_{b_j = 0}\right)\right) \\
d_j | \text{else} &\sim \text{Bernoulli}(p_j) \\
\text{IF } d_j = 1: \quad b_j | \text{else} &\sim N\left(\frac{\mathbf{x}_j'(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}}, \frac{\sigma_e^2}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}}\right) \\
\text{IF } d_j = 0: \quad b_j | \text{else} &\sim N\left(0, \frac{\sigma_e^2}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_{b_j}^{-2}}\right) \\
\sigma_e^2 | \text{else} &\sim \frac{\mathbf{e}' \mathbf{e} + S_e v}{\chi^2(n + v)}
\end{aligned}$$

$$\sigma_{b_j}^2 | \text{else} \sim \frac{b_j^2 + S_b v}{\chi^2(1 + v)}$$

BayesL

$$\begin{aligned} \phi &= (1 - R^2) R^{-2} \sum_j \sigma_{x_j}^2 \\ \mu | \text{else} &\sim N(n^{-1} \sum_i y_i - X_i b, n^{-1} \sigma_e^2) \\ b_j | \text{else} &\sim N\left(\frac{\mathbf{x}_j'(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}_j' \mathbf{x}_j + \lambda_j}, \frac{\sigma_e^2}{\mathbf{x}_j' \mathbf{x}_j + \lambda_j}\right) \\ \sigma_e^2 | \text{else} &\sim \frac{\mathbf{e}' \mathbf{e} + S_e v}{\chi^2(n + v)} \\ \sigma_{b_j}^2 | \text{else} &\sim \frac{b_j^2 + S_b v}{\chi^2(1 + v)} \\ \lambda_j &= \sqrt{\phi \sigma_e^2 \sigma_{b_j}^{-2}} \end{aligned}$$

EM METHODS

emBA

$$\begin{aligned} \mu &= n^{-1} \sum_i y_i - X_i b \\ b_j &= \frac{\mathbf{x}_j'(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_{b_j}^{-2}} \\ \sigma_e^2 &= \frac{\mathbf{e}' \mathbf{e} + S_e v}{n + v} \\ \sigma_{b_j}^2 &= \frac{b_j^2 + S_b v}{1 + v} \end{aligned}$$

emBB

$$\begin{aligned} \mu &= n^{-1} \sum_i y_i - X_i b \\ b_j &= \frac{p(b_j \neq 0)}{p(b_j \neq 0) + p(b_j = 0)} \times \frac{\mathbf{x}_j'(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_{b_j}^{-2}} \\ \sigma_e^2 &= \frac{\mathbf{e}' \mathbf{e} + S_e v}{n + v} \\ \sigma_{b_j}^2 &= \frac{b_j^2 + S_b v}{1 + v} \end{aligned}$$

emBC

$$\begin{aligned} \mu &= n^{-1} \sum_i y_i - X_i b \\ b_j &= \frac{p(b_j \neq 0)}{p(b_j \neq 0) + p(b_j = 0)} \times \frac{\mathbf{x}_j'(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}_j' \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}} \\ \sigma_e^2 &= \frac{\mathbf{e}' \mathbf{e} + S_e v}{n + v} \\ \sigma_b^2 &= \frac{\mathbf{b}' \mathbf{b} + S_b v}{\pi(p + v)} \end{aligned}$$

emBL

$$\begin{aligned}\phi &= (1 - R^2)R^{-2} \sum_j \sigma_{x_j}^2 \\ \mu &= n^{-1} \sum_i y_i - \mathbf{X}_i \mathbf{b} \\ b_j &= 0.5 \frac{|\mathbf{x}'_j(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})| - \alpha \phi}{\mathbf{x}'_j \mathbf{x}_j + (1 - \alpha) \phi} + 0.5 \frac{\mathbf{x}'_j(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{2 \mathbf{x}'_j \mathbf{x}_j}\end{aligned}$$

emRR

$$\begin{aligned}\mu &= n^{-1} \sum_i y_i - \mathbf{X}_i \mathbf{b} \\ b_j &= \frac{\mathbf{x}'_j(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}} \\ \sigma_e^2 &= \frac{\mathbf{e}' \mathbf{e} + S_e v}{n + v} \\ \sigma_b^2 &= \frac{\mathbf{b}' \mathbf{b} + S_b v}{p + v}\end{aligned}$$

emDE

$$\begin{aligned}\phi &= (1 - R^2)R^{-2} \sum_j \sigma_{x_j}^2 \\ \mu &= n^{-1} \sum_i y_i - \mathbf{X}_i \mathbf{b} \\ b_j &= \frac{\mathbf{x}'_j(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2 \sigma_{b_j}^{-2}} \\ \sigma_e^2 &= \frac{\mathbf{e}' \mathbf{y}}{n - 1} \\ \sigma_{b_j}^2 &= b_j^2 + \sigma_e^2 (\mathbf{x}'_j \mathbf{x}_j + \lambda_j)^{-1} \\ \lambda_j &= \sqrt{\phi \sigma_e^2 \sigma_{b_j}^{-2}}\end{aligned}$$

emEN

$$\begin{aligned}\phi &= (1 - R^2)R^{-2} \sum_j \sigma_{x_j}^2 \\ \mu &= n^{-1} \sum_i y_i - \mathbf{X}_i \mathbf{b} \\ b_j &= \frac{|\mathbf{x}'_j(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})| - \alpha \lambda_j}{\mathbf{x}'_j \mathbf{x}_j + (1 - \alpha) \lambda_j} \\ \sigma_e^2 &= \frac{\mathbf{e}' \mathbf{y}}{n - 1} \\ \sigma_b^2 &= b_j^2 + \sigma_e^2 (\mathbf{x}'_j \mathbf{x}_j + \lambda_j)^{-1} \\ \lambda_j &= \sqrt{\phi \sigma_e^2 \sigma_b^{-2}}\end{aligned}$$

emML

$$\begin{aligned}\mu &= n^{-1} \sum_i y_i - \mathbf{X}_i \mathbf{b} \\ b_j &= \frac{\mathbf{x}'_j(\mathbf{y} - 1 \mu - \mathbf{X}_{-j} \mathbf{b}_{-j})}{\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2 \sigma_b^{-2}}\end{aligned}$$

$$\sigma_e^2 = \frac{\mathbf{e}'\mathbf{y}}{n-1}$$

$$\sigma_b^2 = \frac{(\mathbf{y} - 1\mu)'(\mathbf{X}\mathbf{b})}{n \times \sum_j \sigma_{X_j}^2} = \frac{\sigma_y^2 - \sigma_e^2}{\sum_j \sigma_{X_j}^2}$$

mrr (bivariate ridge regression solution for the genomic matrix \mathbf{X})

$$\mu_k = n_k^{-1} \sum_i y_{ik} - \mathbf{X}_{ik} \mathbf{b}_k$$

$$\mathbf{b}_j = \begin{bmatrix} \mathbf{x}'_{j1} \mathbf{x}_{j1} \sigma_{e1}^{-2} + \Sigma^{11} & \Sigma^{12} \\ \Sigma^{21} & \mathbf{x}'_{j2} \mathbf{x}_{j2} \sigma_{e2}^{-2} + \Sigma^{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}'_{j1} (\mathbf{y} - \mathbf{X}_{-j} \mathbf{b}_{-j}) \sigma_{e1}^{-2} \\ \mathbf{x}'_{j2} (\mathbf{y} - \mathbf{X}_{-j} \mathbf{b}_{-j}) \sigma_{e2}^{-2} \end{bmatrix}$$

$$\sigma_b^2(1,2) = \frac{(\mathbf{y}_1 - 1\mu_1)' \mathbf{X}_1 \mathbf{b}_2 + (\mathbf{y}_2 - 1\mu_2)' \mathbf{X}_2 \mathbf{b}_1}{\sum_j \sigma_{X_{j1}}^2 + \sum_j \sigma_{X_{j2}}^2}$$

$$\begin{bmatrix} \Sigma^{11} & \Sigma^{12} \\ \Sigma^{21} & \Sigma^{22} \end{bmatrix} = \begin{bmatrix} \sigma_{b1}^2 & \sigma_{b12} \\ \sigma_{b12} & \sigma_{b2}^2 \end{bmatrix}^{-1}$$

$$\sigma_{ek}^2 = \frac{\mathbf{e}'_k \mathbf{y}_k}{n_k - 1}$$

mkr (bivariate kernel regression solution for kinship matrix \mathbf{K})

$$\mathbf{K} = \mathbf{U}\mathbf{D}\mathbf{U}'$$

$$\mu_k = n_k^{-1} \sum_i y_{ik} - \mathbf{U}_{ik} \mathbf{b}_k$$

$$\mathbf{b}_j = \begin{bmatrix} \mathbf{u}'_{j1} \mathbf{u}_{j1} \sigma_{e1}^{-2} + \Sigma^{11} \mathbf{D}_{jj} & \Sigma^{12} \mathbf{D}_{jj} \\ \Sigma^{21} \mathbf{D}_{jj} & \mathbf{u}'_{j2} \mathbf{u}_{j2} \sigma_{e2}^{-2} + \Sigma^{22} \mathbf{D}_{jj} \end{bmatrix} \begin{bmatrix} \mathbf{u}'_{j1} (\mathbf{y} - 1\mu - \mathbf{U}_{-j} \mathbf{b}_{-j}) \sigma_{e1}^{-2} \\ \mathbf{u}'_{j2} (\mathbf{y} - 1\mu - \mathbf{U}_{-j} \mathbf{b}_{-j}) \sigma_{e2}^{-2} \end{bmatrix}$$

$$\sigma_b^2(1,2) = \frac{(\mathbf{y}_1 - 1\mu_1)' \mathbf{X}_1 \mathbf{b}_2 + (\mathbf{y}_2 - 1\mu_2)' \mathbf{X}_2 \mathbf{b}_1}{\sum_j \sigma_{X_{j1}}^2 + \sum_j \sigma_{X_{j2}}^2}$$

$$\begin{bmatrix} \Sigma^{11} & \Sigma^{12} \\ \Sigma^{21} & \Sigma^{22} \end{bmatrix} = \begin{bmatrix} \sigma_{b1}^2 & \sigma_{b12} \\ \sigma_{b12} & \sigma_{b2}^2 \end{bmatrix}^{-1}$$

$$\sigma_{ek}^2 = \frac{\mathbf{e}'_k \mathbf{y}_k}{n_k - 1}$$

wgr

The wgr is a generalized function that allows users to run some of the whole-genome regression methods with or without subsampling. It also handles missing values.

mixed

The mixed function uses link functions to connect genetic terms and markers. This function is useful to fit linear additive models when the response variables has multiple observations from the same genotypes, allowing to use covariates and other nuisance parameters. Thus:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{Z}(\mathbf{M}\mathbf{a}) + \dots + \mathbf{e}$$

In the example above, the response variable (\mathbf{y}) is a function of fixed effects ($\mathbf{X}\mathbf{b}$), an incidence matrix for genetics (\mathbf{Z}) which connects to the marker matrix (\mathbf{M}) and marker effects (\mathbf{a}), other random terms (...) and residuals (\mathbf{e}). Random effect terms without marker

information (e.g. $\mathbf{Z}_i\mathbf{u}_i$) are referred to as unstructured terms, and their variance components are estimated from the conditional maximum likelihood.

The function can also be used to fit dam-sire models, additive-dominance models, or any models with many random effects. User can select which regression method must be deployed with the argument “alg” (example: alg=emBB), by default the function runs “emML” for a REML-like result. Example of single step model fitted using the mixed function:

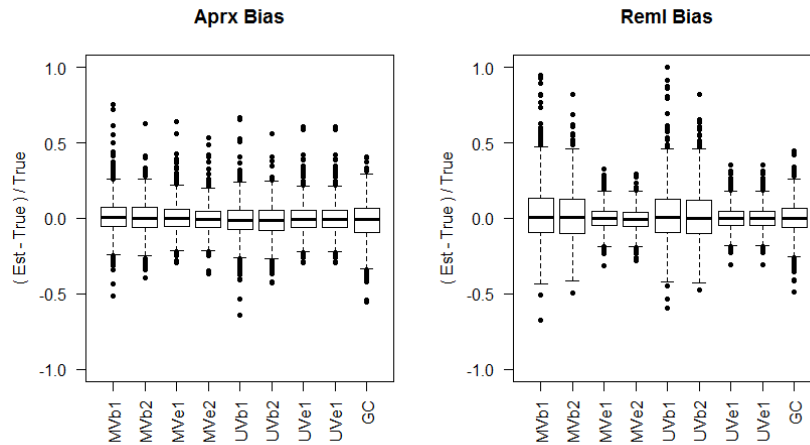
$$\text{Yield} = \text{Environment}(\text{Year}) + \text{Local Contol}(\text{Block}) + \text{Genetics}(\text{ID})$$

```
1 data(met,package='NAM')
2 Gen = as.matrix(Gen)
3 fit = bwGR::mixed(y=YLD,random=~ID+Block,fixed=~Year,data=Obs,X=list(ID=Gen))
```

UNBIASEDNESS

Functions “emML”, “mrr” and “mkr” were implemented with EM-REML like solutions for variance components, based on the pseudo-expectation method provided by Schaeffer (1986). With a series of simulated datasets, we tested the bias of the variance component estimation in a ridge regression model (aka. SNP-BLUP) with varying number of individuals, markers, heritability and genetic correlation.

Below, we present the bias from the pseudo-expectation (Aprx) as implemented in the bwGR package and the exact REML solution (Reml) as implemented in the EMMREML package. The results are presented in the multivariate (M) and univariate (U) settings, evaluating the bias of the genetic variance (Vb), residual variance (Ve) and genetic correlation (GC) between traits 1 and 2 when fitted using multivariate model.



The simulated results indicate that SNP-BLUP fitted via pseudo-expectation yields unbiased variance components, although the sampling variance may differ slightly. Additionally, we observed that the computational time to fit the model SNP-BLUP model using the bwGR package was approximately 8% of the computational cost of EMMREML (data not shown).