Genomic Feature BLUP and Cross Validation

Izel Fourie Sørensen

April 23, 2018

Here we show how to perform a Genomic *Feature* Best Linear Unbiased Prediction (GFBLUP) analyses and cross validation using the greml function (`greml()`) in the qgg package. The GFBLUP model includes an additional genomic effect that quantifies the collective action of a set of markers, i.e. a genomic feature. In this example, the markers that are associated with a chromosome are considered as a feature (set of markers). Performing GFBLUP involves estimating variance components for each of the feature sets with REstricted Maximum Likelihood estimation (REML). The ability of the GFBLUP model to predict the phenotype is assessed using a cross validation procedure. This will be illustrated on the "starvation resistance" phenotype available from the *Drosophila melanogaster* Genetic Reference Panel (DGRP). For more information on the DGRP go to the website: http://dgrp2.gnets.ncsu.edu/.

To perform a GFBLUP analysis the following input data is essential.

1. $\boldsymbol{y}$: vector of phenotype
2. $\boldsymbol{X}$: design matrix for covariables
3. Genomic feature marker sets
4. $\boldsymbol{W}$: centered and scaled genotype matrix
5. $\boldsymbol{G}$: genomic relationship matrix for each feature

This script includes the following steps for performing a GFBLUP analysis and cross validation: 1) load and prepare data for GFBLUP analysis, 2) restricted maximum likelihood (REML) analyses for estimating variance components, and 3) REML/GFBLUP analysis and cross validation.

Install the qgg package.

```
library(devtools)
install_github("psoerensen/qgg")
```

```
library(qgg)
```

Load and prepare data for GFBLUP analysis

Load the cleaned and edited data frame (prepared earlier) of the starvation resistance data.

```
load(file = "./phenotypes/starv_inv_wo.Rdata")
dim(starv)
```

```
## [1] 406  20
```

```
head(starv)
```

```
##     L sex        y In_2L_t In_2R_NS In_2R_Y1 In_2R_Y2 In_2R_Y3 In_2R_Y4
## 1 100   M 49.28000  INV/ST       ST       ST       ST       ST       ST
## 2 101   M 47.20000  INV/ST       ST       ST       ST       ST       ST
## 3 105   M 51.04000      ST       ST       ST       ST       ST       ST
## 4 109   M 44.96000  INV/ST       ST       ST       ST       ST       ST
## 5 129   M 33.08475      ST       ST       ST       ST       ST       ST
## 6 136   M 63.04000      ST       ST       ST       ST       ST       ST
##   In_2R_Y5 In_2R_Y6 In_2R_Y7 In_3L_P In_3L_M In_3L_Y In_3R_P In_3R_K
## 1       ST       ST       ST      ST      ST      ST      ST     INV
## 2       ST       ST       ST      ST      ST      ST      ST      ST
## 3       ST       ST       ST      ST      ST      ST      ST     INV
```

```
## 4       ST       ST       ST      ST      ST      ST      ST      ST
## 5       ST       ST       ST      ST      ST      ST      ST      ST
## 6       ST       ST       ST  INV/ST      ST      ST      ST  INV/ST
##   In_3R_Mo In_3R_C wo
## 1       ST      ST  y
## 2       ST      ST  n
## 3       ST      ST  n
## 4       ST      ST  n
## 5       ST      ST  n
## 6       ST      ST  y
```

Create a vector of the starvation resistance phenotype, $\boldsymbol{y}$.

```
data <- starv
y <- data$y
length(y)
```

```
## [1] 406
```

Prepare the design matrix $\boldsymbol{X}$ for the covariables. `fm` is the formula used for including the relevant covariables in the design matrix.

```
fm <- y ~  wo + In_2L_t + In_2R_NS + In_3R_P + In_3R_K + In_3R_Mo
X <- model.matrix(fm, data=data)
X[1:5,]
```

```
##   (Intercept) woy In_2L_tINV/ST In_2L_tST In_2R_NSINV/ST In_2R_NSST
## 1           1   1             1         0              0          1
## 2           1   0             1         0              0          1
## 3           1   0             0         1              0          1
## 4           1   0             1         0              0          1
## 5           1   0             0         1              0          1
##   In_3R_PINV/ST In_3R_PST In_3R_KINV/ST In_3R_KST In_3R_MoINV/ST
## 1             0         1             0         0              0
## 2             0         1             0         1              0
## 3             0         1             0         0              0
## 4             0         1             0         1              0
## 5             0         1             0         1              0
##   In_3R_MoST
## 1          1
## 2          1
## 3          1
## 4          1
## 5          1
```

Load the centered and scaled genotype matrix, $\boldsymbol{W}$ created earlier.

```
load(file = "./genotypes/dgrp2_W2.Rdata")
dim(W)
```

```
## [1]     205 1725755
```

```
W[1:5,1:5]
```

```
##       2L_5317    2L_5372    2L_5390    2L_5403    2L_5465
## 21 -0.2486289 -0.6646914  1.2122772 -0.4059216 -0.4007831
## 26 -0.2486289 -0.6646914 -0.8200699 -0.4059216 -0.4007831
## 28 -0.2486289 -0.6646914  1.2122772 -0.4059216 -0.4007831
```

```
## 31  4.0006648 -0.6646914 -0.8200699 -0.4059216 -0.4007831
## 32 -0.2486289 -0.6646914 -0.8200699 -0.4059216 -0.4007831
```

Load the chromosome marker sets as created earlier (under annotation).

Markers in this dataset are distributed on 6 chromosome arms, i.e., 2L, 2R, 3L, 3R, 4 and X. Each chromosome and its associated markers is considered a marker set (genomic feature, `GF`) in this example.

```
load(file = "./annotation/chrSets2.Rdata")
str(chrSets)
```

```
## List of 6
##  $ 2L: chr [1:406577] "2L_5317" "2L_5372" "2L_5390" "2L_5403" ...
##  $ 2R: chr [1:327967] "2R_10037" "2R_10468" "2R_10959" "2R_12079" ...
##  $ 3L: chr [1:390711] "3L_39998" "3L_40145" "3L_40202" "3L_40635" ...
##  $ 3R: chr [1:368096] "3R_1339" "3R_1651" "3R_2158" "3R_2318" ...
##  $ 4 : chr [1:2686] "4_61790" "4_62622" "4_62905" "4_62908" ...
##  $ X : chr [1:229718] "X_19380" "X_19797" "X_20390" "X_20491" ...
```

Keep only SNPs that are in the W matrix.

```
setsGF <- lapply(chrSets,function(x){ x[x%in%colnames(W)] })
```

Create the objects `nsets`: the number of sets (the number of chromosomes) and `nsnps`: the number of SNPs per chromosome.

```
nsets <- length(setsGF)
nsnps <- sapply(setsGF,length)
```

The additive genomic relationship matrix $\boldsymbol{G}$ (VanRaden PM. 2008. J Dairy Sci. 91:4414-4423) is constructed using all genetic markers as follows: $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{W}'/m$, where $\boldsymbol{W}$ is the centered and scaled genotype matrix, and m is the total number of markers. The *genomic feature* relationship matrix (`GF`) $\boldsymbol{G}_k = \boldsymbol{W}_k\boldsymbol{W}_k'/m_k$, is the additive genomic relationship matrix for the k$^{\text{th}}$ genetic marker set.

Compute the additive genomic relationship matrix, $\boldsymbol{G}$, from the genotype matrix, $\boldsymbol{W}$.

```
L <- data$L
GF <- lapply(setsGF, function(x) {computeGRM(W = W[L, x])})
```

Starvation resistance was measured for males and females for each line. Therefore GF has a dimension of 406 rows and 406 columns for each chromosome.

```
str(GF)
```

```
## List of 6
##  $ 2L: num [1:406, 1:406] 0.50604 0.09577 -0.00744 0.11721 -0.03955 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:406] "100" "101" "105" "109" ...
##   .. ..$ : chr [1:406] "100" "101" "105" "109" ...
##  $ 2R: num [1:406, 1:406] 0.997155 -0.000163 -0.000395 -0.005106 -0.005151 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:406] "100" "101" "105" "109" ...
##   .. ..$ : chr [1:406] "100" "101" "105" "109" ...
##  $ 3L: num [1:406, 1:406] 1.05858 -0.00874 0.01605 0.00636 0.00736 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:406] "100" "101" "105" "109" ...
##   .. ..$ : chr [1:406] "100" "101" "105" "109" ...
```

```
##  $ 3R: num [1:406, 1:406] 1.42308 -0.01439 0.35384 -0.00985 -0.02452 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:406] "100" "101" "105" "109" ...
##   .. ..$ : chr [1:406] "100" "101" "105" "109" ...
##  $ 4 : num [1:406, 1:406] 0.6 -0.254 -0.319 0.239 0.108 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:406] "100" "101" "105" "109" ...
##   .. ..$ : chr [1:406] "100" "101" "105" "109" ...
##  $ X : num [1:406, 1:406] 1.01872 0.00499 0.0278 0.00232 -0.01235 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:406] "100" "101" "105" "109" ...
##   .. ..$ : chr [1:406] "100" "101" "105" "109" ...
```

Restricted maximum likelihood (REML) analyses

Behind the scenes of the greml function:
REML analyses are used for estimating the variance components, $\sigma^2_{g_k}$ (variance component for the k$^{\text{th}}$ feature) and σ^2_e for the random effects in the linear mixed model:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \sum_{k=1}^{n_f} \boldsymbol{Z}\boldsymbol{g}_k + \boldsymbol{e}$$

where $\boldsymbol{y}$ is the vector of phenotypic observations, $\boldsymbol{X}$ and $\boldsymbol{Z}$ are design matrices for the fixed and random effects, $\boldsymbol{b}$ is a vector of fixed effects, $\boldsymbol{g}_k$ is the vector of genetic values captured by the genetic markers in the k$^{\text{th}}$ feature, $\boldsymbol{e}$ is the vector of residuals, and n_f the number of features. The random genomic values and the residuals were assumed to be independent normally distributed values described as follows: $\boldsymbol{g}_k \sim N(0, \boldsymbol{G}_k\sigma^2_{g_k})$ and $\boldsymbol{e} \sim N(0, \boldsymbol{I}\sigma^2_e)$. Thus, we assume that the observed phenotypes $\boldsymbol{y} \sim N(\boldsymbol{X}\boldsymbol{b}, V)$ where $\boldsymbol{V} = \sum_{k=1}^{n_f} \boldsymbol{Z}\boldsymbol{G}_k\boldsymbol{Z}'\sigma^2_{g_k} + \boldsymbol{I}\sigma^2_e$.

Conditional on the observed phenotype in the study population the genetic predisposition associated with the k$^{\text{th}}$ marker set can be computed as:

$$\hat{g_k} = \hat{\sigma}^2_{g_k} G_k Z' \hat{V}^{-1}(y - X\hat{b})$$

The phenotype is predicted as:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{b}} + \sum_{k=1}^{n_f} \boldsymbol{Z}\hat{\boldsymbol{g}}_k$$

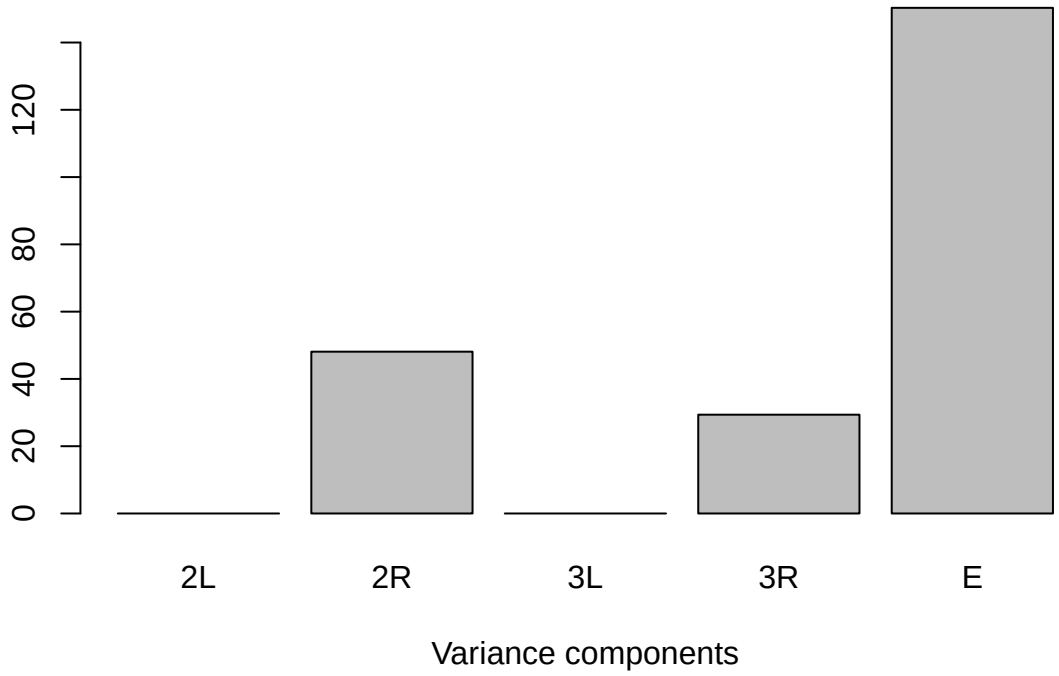
The `greml` function goes through a number of iterations before convergence (i.e., the change in parameters between consecutive rounds becomes smaller than a specified threshold, see "tol" argument in the greml help page). In this example the four chromosome arms, 2L, 2R, 3L and 3R, are included as feature sets. Each iteration returns values for the variance components, $\sigma^2_{g_{2L}}$ (third column), $\sigma^2_{g_{2R}}$ (fourth column), $\sigma^2_{g_{3L}}$ (fifth column), $\sigma^2_{g_{3R}}$ (sixth column), and σ^2_e (seventh column). In this case the variance components for chromosome arms 2L and 3L converged to 0.

The `greml` function returns a list structure that includes estimates of the fixed effects ($\hat{b}$), random effects ($\hat{g}_k$), residual effects ($\hat{e}$), as well as the variance components ($\sigma^2_{g_k}$). Other values in the list are described on the `greml` help page. Change `verbose = FALSE` to `verbose = TRUE` to see the iterations.

```
fitGF <- greml(y=y, X=X, G=GF[1:4], verbose = FALSE)
```

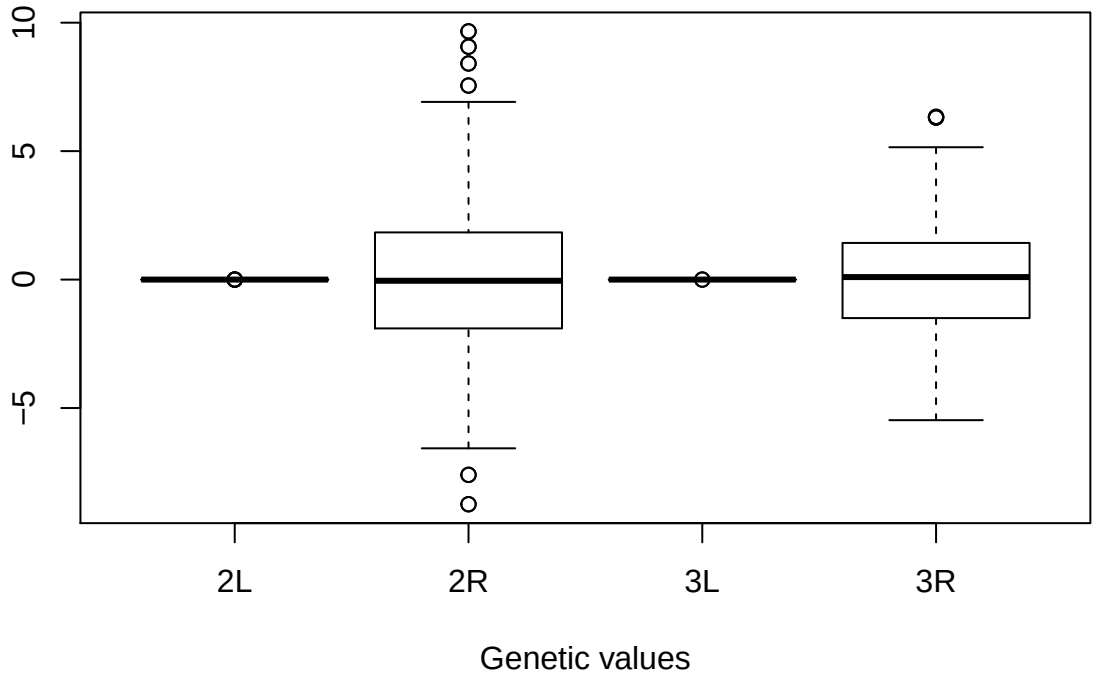
The variance components estimated for chromosome arms 2L and 3L are close to zero.

```r
barplot(as.vector(fitGF$theta), xlab = "Variance components",
  names.arg = c("2L", "2R", "3L", "3R", "E"))
```



As expected (based on the small variance components for 2L and 3L), the genomic variance is extremely small for chromosome arms 2L and 3L. Therefore, it makes sence not to include 2L and 3L for predicting y.

```r
boxplot(fitGF$g, xlab = "Genetic values", names = c("2L", "2R", "3L", "3R"))
```



3. REML analyses and cross validation using the GFBLUP model

The genomic parameters are estimated for the phenotypes from the lines in the training data (t,365 lines) and the phenotype is predicted for the lines in the validation data (v, 41 lines). The genetic value is predicted based on the genomic relationship between the training and validation population:

$$\hat{g}_k^v = \hat{\sigma}^2_{g_k} G_k^{vt} Z' \hat{V}_t^{-1} (y_t - X_t \hat{b_t})$$

Where the genomic relationship matrix

$$G_k = \begin{pmatrix} G_k^{vv} & G_k^{vt} \\ G_k^{vt} & G_k^{tt} \end{pmatrix}$$

is partitioned according to relationships between the individuals in the training data G_k^{tt}, between the individuals in the validation data G_k^{vv} and between the individuals in the training and validation data G_k^{vt}. Thus the total genomic predisposition is predicted using the estimated variance components ($\sigma^2_{g1}, ..., \sigma^2_{n_f}$ and σ^2_e) in the training data. The right-most term, $(y_t - X_t \hat{b_t})$, constitutes the phenotypes corrected for fixed effects for the individuals in the training data. The inverse term $\hat{V}_t^{-1}$ is essentially the variance-covariance structure for the corrected phenotypes. These two terms multiplied together are the standardized and corrected phenotypes for the individuals in the training data, which are projected onto the total genetic covariance structure between the training and the validation data.

Based on the variance components estimated for the *training set*, the phenotype in the *validation set* is predicted as:

$$\hat{y_v} = X_v \hat{b_t} + \sum_{k=1}^{n_f} Z \hat{g}_k^v$$

5 validation sets are created by randomly sampling 41 values from 1 - 406 (the length of y), and repeating this sampling 5 times. The validation sets are saved in the `validate` matrix. This matrix specifies the rows of the data to be used in the GREML analyses.

Since chrosomes 2L and 3L have variance components of 0, only chromosomes 2R and 3R are included as features for the prediction of y.

```
n <- length(y)
validate <- replicate(5, sample(1:n, 41))
cvG <- greml(y = y, X = X, G = GF[c(2, 4)], validate = validate)

library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.4.4
```

```
kable(cvG$pred, caption = "Cross Validation Predictive Ability")
```

Table 1: Cross Validation Predictive Ability

Corr	R2	Nagel R2	AUC	intercept	slope	MSPE
-0.082	0.007	NA	NA	65.753	-0.231	146.528
0.224	0.050	NA	NA	15.271	0.767	206.006
0.197	0.039	NA	NA	23.434	0.660	198.266
0.247	0.061	NA	NA	16.945	0.631	126.951
0.077	0.006	NA	NA	31.755	0.424	259.266

```
kable(cvG$theta, caption = "Cross Validation Parameter Estimates")
```

Table 2: Cross Validation Parameter Estimates

2R	3R	E
13.102	22.158	147.605
18.739	8.654	146.194
12.430	20.238	143.759
23.293	12.677	148.891
22.279	6.140	140.735

Output of the GREML cross validation analysis

The output includes statistics that quantify the model's predictive ability as assessed by regressing the observed phenotype against the predicted phenotype for the validation data set: $y = intercept + \hat{y} slope + e$

Value	Description
Corr	Correlation between the predicted and observed phenotypic value. Averaging the list of 5 correlations yields the predictive ability
R2	R^2, proportion of the total variance that is explained by the GFBLUP model
Nagel R2	Nagelkerke's R
AUC	Area Under the ROC Curve
intercept	The y-axis intercept for the regression of y unto $\hat{y}$
slope	Slope for the regression of y unto $\hat{y}$
MSPE	mean squared prediction error $= \frac{1}{n_v}\sum_{i=1}^{n_v}(y_i - \hat{y}_i)^2$, n_v = number of observations in validation set
2R	Estimated variance component, $\hat{\sigma}^2_{g_{2R}}$

Value	Description
3R	Estimated variance component, $\hat{\sigma}^2_{g_{3R}}$
E	Estimated variance component, $\hat{\sigma}^2_e$

Prepare data frame of results

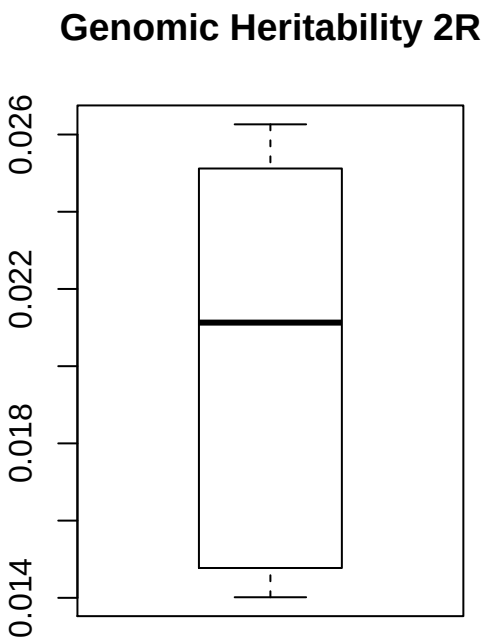
```
cvG_mean <- round(colMeans(cvG$pred), digits = 3)

cvGsem <- apply(cvG$pred, 2, function(x){sd(x)/sqrt(5)})
cvG_sem <- round(cvGsem, digits = 3)

results <- data.frame(rbind(cvG_mean,cvG_sem))
kable(results, caption = "Results Summary")
```

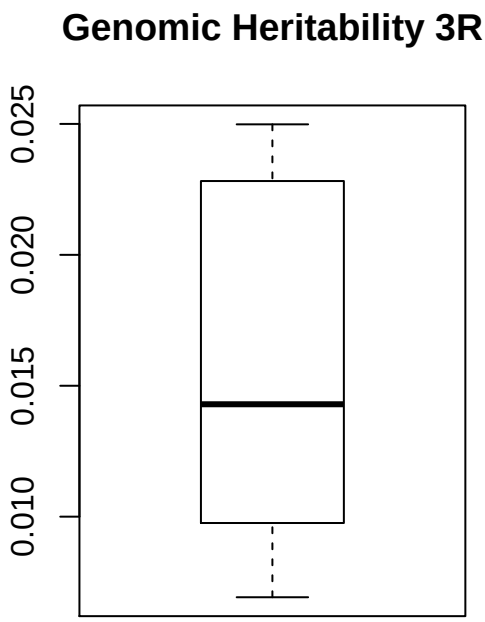
Table 4: Results Summary

	Corr	R2	Nagel.R2	AUC	intercept	slope	MSPE
cvG_mean	0.133	0.033	NA	NA	30.632	0.450	187.403
cvG_sem	0.061	0.011	NA	NA	9.246	0.179	23.403

Genomic heritability ($\hat{h}^2$)

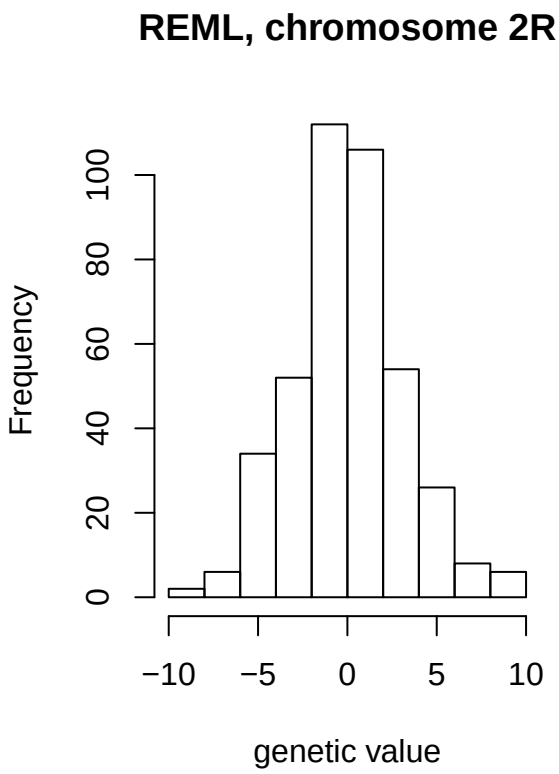
```
theta <- cvG$theta
h2f2 <- theta["2R"]/sum(theta)
h2f4 <- theta["3R"]/sum(theta)

layout(matrix(1:2,ncol=2))
boxplot(h2f2, main = "Genomic Heritability 2R")
boxplot(h2f4, main = "Genomic Heritability 3R")
```

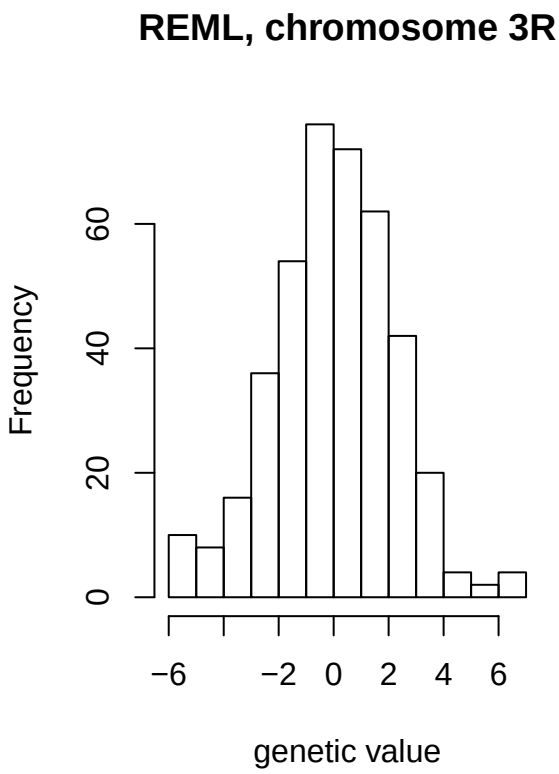




Histogram of genetic values

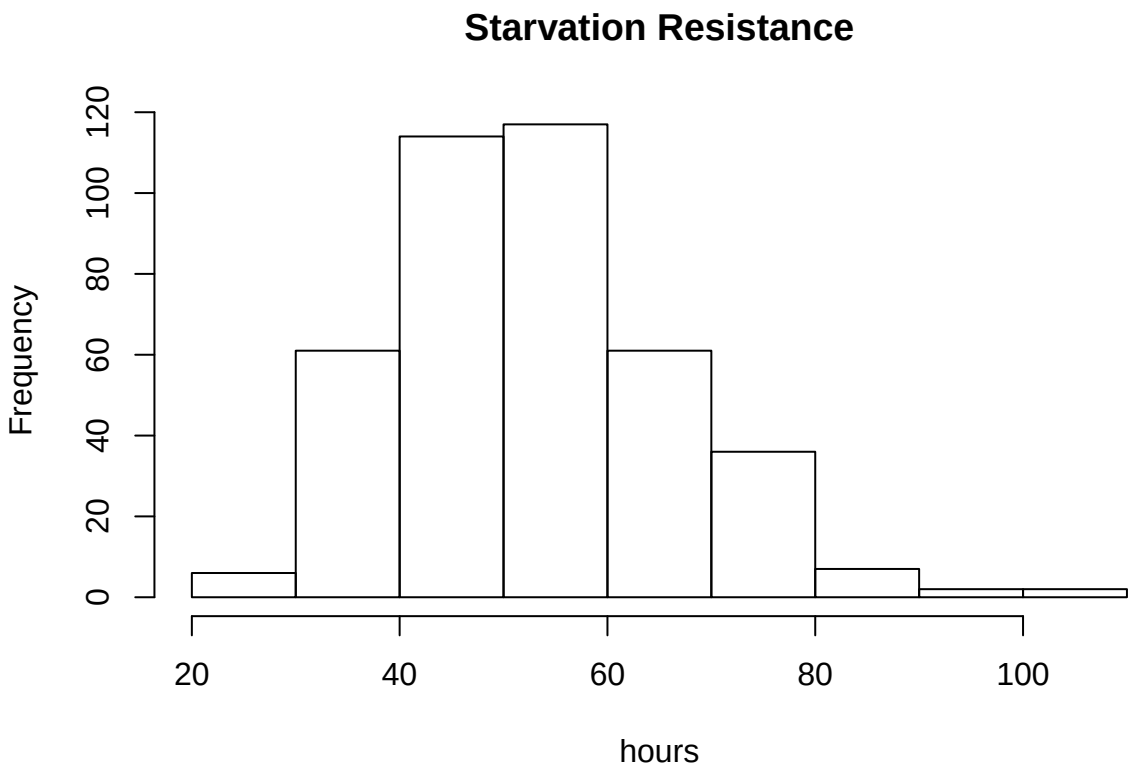
```
layout(matrix(1:2,ncol=2))
hist(fitGF$g[,2], xlab = "genetic value", main = "REML, chromosome 2R")
hist(fitGF$g[,4], xlab = "genetic value", main = "REML, chromosome 3R")
```





Histogram of phenotype

```
hist(y, xlab = "hours", main = "Starvation Resistance")
```


Starvation Resistance
Frequency
0
20
40
60
80
100
120
20
40
60
80
100
hours