File S1

Supplementary Materials

A Prior Densities Used in the BGLR R-Package

In this section we describe the prior distributions assigned to the location parameters, $(\boldsymbol{\beta}_j, \boldsymbol{u}_l)$, entering in the linear predictor of eq. (1). For each of the unknown effects included in the linear predictor, $\{\boldsymbol{\beta}_1, .., \boldsymbol{\beta}_J, \boldsymbol{u}_1, ..., \boldsymbol{u}_L\}$, the prior density assigned is specified via the argument `model` in the corresponding entry of the list (see Box 8 for an example). Table S1 describes, for each of the options implemented, the prior density used. A brief description is given below.

FIXED. In this case regression coefficients are assigned flat priors, specifically we use a Gaussian prior with mean zero and variance equal to 1×10^{10}.

BRR. When this option is used regression coefficients are assigned normal IID normal distributions, with mean zero and variance σ^2_β. In a 2nd level of the hierarchy, the variance parameter is assigned a scaled-inverse Chi-squared density, with parameters df_β and S_β. This density is parameterized in a way that the prior expected value and mode are $E(\sigma^2_\beta) = \frac{S_\beta}{df_\beta - 2}$ and $Mode(\sigma^2_\beta) = \frac{S_\beta}{df_\beta + 2}$, respectively. By default, if df_β and S_β are not provided, BGLR sets $df_\beta = 5$ and solves for the scale parameter to match the R-squared of the model (see default rules to set hyper-parameters below). An analysis with fixed variance parameter can be obtained by choosing the degree of freedom parameter to a very large value (e.g., 1×10^{10}) and solving for the scale using $S_\beta = \sigma^2_\beta \times (df_\beta + 2)$; this gives a prior that collapses to a point of mass at σ^2_β.

BayesA. In this model the marginal distribution of marker effects is a scaled-t density, with parameters df_β and S_β. For computational convenience this density is implemented as an infinite mixture of scaled-normal densities. In a first level of the hierarchy marker effects are assigned normal densities with zero mean and marker-specific variance parameters, $\sigma^2_{\beta_{jk}}$. In a 2nd level of the hierarchy these variance parameters are assigned IID scaled-inverse Chi-squared densities with degree of freedom and scale parameters df_β and S_β, respectively. The degree of freedom parameter is regarded as known; if the user does not provide a value for this parameter BGLR sets $df_\beta = 5$. The scale parameter is treated as unknown, and BGLR assigns to this parameter a gamma density with rate and shape parameters r and s, respectively. The mode and coefficient of variation (CV) of the gamma density are $Mode(S_\beta) = (s-1)/r$ (for $s > 1$) and $CV(S_0) = 1/\sqrt{s}$. If the user does not provide shape and rate parameters BGLR sets $s = 1.1$, this gives a relatively un-informative prior with a CV of approximately 95%, and then solves for the rate so that the total contribution of the linear predictor matches the R-squared of the model (see default rules to set hyper-parameters, below). If one wants to run the analysis with fixed scale one can choose a very large value for the shape parameter (e.g., 1×10^{10}) and then solve for the rate so that the prior mode matches the desired value of the scale parameter using $r = (s-1)/S_\beta$.

Bayesian LASSO (BL). In this model the marginal distribution of marker effects is double-exponential. Following Park and Casella (2008) we implement the double-exponential density as a mixture of scaled normal densities. In the first level of the hierarchy, marker effects are assigned independent normal densities with null mean and maker-specific variance parameter $\tau^2_{jk} \times \sigma^2_\varepsilon$. The residual variance is assigned a scaled-inverse Chi-square density, and the marker-specific scale parameters, τ_{jk}^2, are assigned IID exponential densities with rate parameter $\lambda^2/2$. Finally, in the last level of the hierarchy λ^2 is either regarded as fixed (this is obtained by setting in the linear predictor the option `type="FIXED"`), or assigned either a Gamma ($\lambda^2 \sim Gamma(r, s)$ if `type="gamma"`) or a $\lambda/\max$ is assigned a Beta prior, if `type="beta"`, here max is a user-defined parameter representing the maximum value that λ can take). If nothing is specified, BGLR sets `type="gamma"` and $s = 1.1$, and solves for the scale parameter to match the expected R-squared of the model (see section B of this Supplementary Materials for further details).

BayesB-C. In these models marker effects are assigned IID priors that are mixtures of a point of mass at zero and a slab that is either normal (BayesC) or a scaled-t density (BayesB). The slab is structured as either in the BRR (this is the case of BayesC) or as in BayesA (this is the case of BayesB). Therefore, BayesB and BayesC extend BayesA and BRR, respectively, by introducing an additional parameter π which in the case of BGLR represents the prior proportion of non-zero effects. This parameter is treated as unknown and it is assigned a Beta prior $\pi \sim Beta(p_0, \pi_0)$, with $p_0 > 0$ and $\pi_0 \in [0, 1]$. The beta prior is parameterized in a way that the expected value by $E(\pi) = \pi_0$; on the other hand p_0 can be interpreted as the number of prior counts (priors “successes” plus prior “failures”); with this parametrization the variance of the Beta distribution is then given by $Var(\pi) = \frac{\pi_0(1-\pi_0)}{(p_0+1)}$, which is inversely proportional to p_0. Choosing $p_0 = 2$ and $\pi_0 = 0.5$ gives a uniform prior in the interval $[0, 1]$. Choosing a very large value for p_0 gives a prior that collapses to a point of mass at π_0.

Table S1. Prior densities implemented in BGLR.

`model=`	**Join distribution of effects and hyper-parameters**	**Specification of elements in the linear predictor**
`FIXED`	$p(\boldsymbol{\beta}_j) \propto 1$	`list(X=, model="FIXED")`
`BRR`	$p(\boldsymbol{\beta}_j, \sigma^2_\beta) = \left\{\prod_k N(\beta_{jk}\|0, \sigma^2_\beta)\right\} \chi^{-2}(\sigma^2_\beta\|df_\beta, S_\beta)$	`list(X=, model="BRR",df0=,S0=,R2=)`
`BayesA`	$p(\boldsymbol{\beta}_j, \sigma^2_{\beta_j}, S_\beta) = \left\{\prod_k N(\beta_{jk}\|0, \sigma^2_{\beta_{jk}})\chi^{-2}(\sigma^2_{\beta_{jk}}\|df_\beta, S_\beta)\right\} G(S_\beta\|r, s)$	`list(X=, model="BayesA",df0=,rate0=, shape0=,R2=)`
`BL`	$p(\boldsymbol{\beta}_j, \boldsymbol{\tau}^2_j, \lambda^2\|\sigma^2_\varepsilon) = \left\{\prod_k N(\beta_{jk}\|0, \tau^2_{jk} \times \sigma^2_\varepsilon) Exp\left\{\tau^2_{jk}\|\frac{\lambda^2}{2}\right\}\right\} \times G(\lambda^2\|r, s)$, or	`list(X=,model="BL",lambda=,type="gamma", rate=,shape=,R2=)`[1]
	$p(\boldsymbol{\beta}_j, \boldsymbol{\tau}^2_j, \lambda\|\sigma^2_\varepsilon, \texttt{max}) = \left\{\prod_k N(\beta_{jk}\|0, \tau^2_{jk} \times \sigma^2_\varepsilon) Exp\left\{\tau^2_{jk}\|\frac{\lambda^2}{2}\right\}\right\} \times B(\lambda/\max\|p_0, \pi_0)$, or	`list(X=,model="BL",lambda=,type="beta", probIn=,counts=,max=,R2=)`[1]
	$p(\boldsymbol{\beta}_j, \boldsymbol{\tau}^2_j\|\sigma^2_\varepsilon, \lambda) = \left\{\prod_k N(\beta_{jk}\|0, \tau^2_{jk} \times \sigma^2_\varepsilon) Exp\left\{\tau^2_{jk}\|\frac{\lambda^2}{2}\right\}\right\}$	`list(X=,model="BL",lambda=,type="FIXED")`[1]
`BayesC`	$p(\boldsymbol{\beta}_j, \sigma^2_\beta, \pi) = \left\{\prod_k \left[\pi N(\beta_{jk}\|0, \sigma^2_\beta) + (1-\pi)1(\beta_{jk}=0)\right]\right\} \times\chi^{-2}(\sigma^2_\beta\|df_\beta, S_\beta) B(\pi\|p_0, \pi_0)$	`list(X=,model="BayesC",df0,S0, probIn=,counts=,R2=)`[2]
`BayesB`	$p(\boldsymbol{\beta}_j, \sigma^2_\beta, \pi) = \left\{\prod_k \left[\pi N(\beta_{jk}\|0, \sigma^2_\beta) + (1-\pi)1(\beta_{jk}=0)\right] \chi^{-2}(\sigma^2_{\beta_{jk}}\|df_\beta, S_\beta)\right\} B(\pi\|p_0, \pi_0) \times G(S_\beta\|r, s)$	`list(X=,model="BayesB",df0,rate0,shape0, probIn=,counts=,R2=)`[2]
`RKHS`	$p(\boldsymbol{u}_l, \sigma^2_{u_l}) = N(\boldsymbol{u}_l\|\mathbf{0}, \boldsymbol{K}_l \times \sigma^2_{u_l})\chi^{-2}(\sigma^2_{u_l}\|df_l, S_l)$	Either `list(K=,model="RKHS",df0,S0,R2=)` or `list(V=,d=,model="RKHS",df0,S0,R2=)`[3]

$N(\cdot|\cdot,\cdot)$, $\chi^{-2}(\cdot|\cdot,\cdot)$, $G(\cdot|\cdot,\cdot)$, $Exp(\cdot|\cdot)$, $B(\cdot|\cdot,\cdot)$ denote normal, scaled inverse Chi-squared, gamma, exponential and beta densities, respectively. (1) `type` can take values `"FIXED"`, `"gamma"`, or `"beta"`; (2) `probIn` represents the prior probability of a marker having a non-null effect (π_0), counts (the number of 'prior counts') can be used to control how informative the prior is; (3) `V` and `d` represent the eigen-vectors and eigen-values of $\boldsymbol{K}$, respectively.

B Default rules for choosing hyper-parameters

BGLR has built-in rules to set values of hyper-parameters. The default rules assign proper, but weakly informative, priors with prior modes chosen in a way that, a priori, they obey a variance partition of the phenotype into components attributable to the error terms and to each of the elements of the linear predictor. The user can control this variance partition by setting the argument R2 (representing the **model R-squared**) of the BGLR function to the desired value. By default the model R2 is set equal to 0.5, in which case hyper-parameters are chosen to match a variance partition where 50% of the variance of the response is attributable to the linear predictor and 50% to model residuals. Each of the **elements of the linear predictor** has its own R2 parameter (see last column of Table S1). If these are not provided, the **R2** attributable to each element of the linear predictor equals the R-squared of the model divided the number of elements in the linear predictor. Once the R2 parameters are set, BGLR checks whether each of the hyper-parameters have been specified and if not, the built in-rules are used to set values for these hyper-parameters. Next we briefly describe the built-in rules implemented in BGLR; these are based on formulas similar to those described by de los Campos et al. (2013) implemented using the prior mode instead of the prior mean.

Variance parameters. The residual variance (σ_ε^2), $\sigma_{u_l}^2$ of the RKHS model, and σ_β^2, of the BRR, are assigned scaled-inverse Chi-square densities, which are indexed by a **scale** and a **degree of freedom** parameter. By default, if degree of freedom parameter is not specified, these are set equal to 5 (this gives a relatively un-informative scaled-inverse Chi-square and guarantees a finite prior variance) and the scale parameter is solved for to match the desired variance partition. For instance, in case of the residual variance the scale is calculated using $S_\varepsilon = var(y) \times (1 - R2) \times (df_\varepsilon + 2)$, this gives a prior mode for the residual variance equal to $var(y) \times (1 - R2)$. Similar rules are used in case of other variance parameters. For instance, if one element of the linear predictor involves a linear regression of the form $\boldsymbol{X\beta}$ with `model='BRR'` then $S_\beta = var(y) \times R2 \times (df_\beta + 2)/MSx$ where MSx is the sum of the sample variances of the columns of $\boldsymbol{X}$ and R2 is the proportion of phenotypic variance a-priori assigned to that particular element of the linear predictor. The selection of the scale parameter when the model is the RKHS regression is modified relative to the above rule to account for the fact that the average diagonal value of $\boldsymbol{K}$ may be different than 1, specifically we choose the scale parameter according to the following formula $S_l = var(y) \times R2 \times (df_l + 2)/mean(diag(\boldsymbol{K}))$.

In models **BayesA** and **BayesB** the scale-parameter indexing the t-prior assigned to marker effects is assigned a Gamma density with rate and shape parameters r and s, respectively. By default BGLR sets $s = 1.1$ and solves for the rate parameter using $r = (s-1)/S_\beta$ with $S_\beta = var(y) \times R2 \times (df_\beta + 2)/MSx$, here, as before, MSx represents the sum of the variances of the columns of X.

For the **BL**, the default is to set: `type='gamma'`, fix the shape parameter of the gamma density assigned λ^2 to 1.1 and then solve for the rate parameter to match the expected proportion of variance accounted for by the corresponding element of the linear predictor, as specified by the argument R2. Specifically, we set the rate to be $r = (s - 1)/(2 \times (1 - R2)/R2 \times MSx)$.

For models **BayesB** and **BayesC**, the default rule is to set $\pi_0 = 0.5$ and $p_0 = 10$. This gives a weakly informative beta prior for π with a prior mode at 0.5. The scale and degree-of freedom parameters entering in the priors of these two models are treated as in the case often models

BayesA (in the case of BayesB) and BRR (in the case of BayesC), but the rules are modified by considering that, a-priori, only a fraction of the markers (π) nave non-null effects; therefore, in BayesC we use $S_\beta = var(y) \times R2 \times (df_\beta + 2)/MSx/\pi$ and in BayesB we set $r = (s-1)/S_\beta$ with $S_\beta = var(y) \times R2 \times (df_\beta + 2)/MSx/\pi$.

C Supplementary R scripts

Box S1 illustrates how to extract estimates and predictions form the models fitted in Box 7.

Box S1: Supplementary code for the model fitted in Box 7

```
#Residual Variance
 fmBRR$varE; fmBRR$SD.varE
 fmBA$varE; fmBA$SD.varE
 fmBB$varE; fmBB$SD.varE
# DIC and pD
 fmBRR$fit
 fmBA$fit
 fmBB$fit
#Predictions
 fmBRR$yHat; fmBRR$SD.yHat
 fmBA$yHat;  fmBA$SD.yHat
 fmBB$yHat;  fmBB$SD.yHat
#Correlations between predicted and simulated signals
 cor(signal,fmBRR$yHat)
 cor(signal,fmBA$yHat)
 cor(signal,fmBB$yHat)

# Estimated effects
 tmp<-range(abs(b0))
 plot(numeric()~numeric(),ylim=tmp,xlim=c(1,p),
      ylab=expression(paste("|",beta[j],"|")),
      xlab="Marker Possition (order)")
 abline(v=whichQTL,lty=2,col=4)
 points(x=whichQTL,y=abs(b0[whichQTL]),pch=19,col=4)
 points(x=1:p,y=abs(fmBRR$ETA$MRK$b),col=1,cex=.5)
 lines(x=1:p,y=abs(fmBRR$ETA$MRK$b),col=1,cex=.5)
 points(x=1:p,y=abs(fmBB$ETA$MRK$b),col=2,cex=.5)
 lines(x=1:p,y=abs(fmBB$ETA$MRK$b),col=2,cex=.5)
```

Box S2 illustrates how to extract estimates and predictions form the models fitted in Box 8.

Box S2: Supplementary code for the model fitted in Box 8

```
#1# Estimated Marker Effects & posterior SDs
  bHat<- fm$ETA$MRK$b
  SD.bHat<- fm$ETA$MRK$SD.b
  plot(bHat^2, ylab="Estimated Squared-Marker Effect",
        type="o",cex=.5,col="red",main="Marker Effects",
        xlab="Marker")
   points(bHat^2,cex=0.5,col="blue")

#2# Predictions
  # Genomic Prediction
    gHat<-X%*%fm$ETA$MRK$b
    plot(fm$y~gHat,ylab="Phenotype",
         xlab="Predicted Genomic Value", col=2, cex=0.5,
         main="Predicted Genomic Values Vs Phenotypes",
         xlim=range(gHat),ylim=range(fm$y));

#3# Godness of fit and related statistics
   fm$fit
   fm$varE # compare to var(y)

#4# Trace plots
  list.files()

  # Residual variance
   varE<-scan("varE.dat")
   plot(varE,type="o",col=2,cex=.5,
          ylab=expression(sigma[epsilon]^2),
          xlab="Sample",main="Residual Variance");
   abline(h=fm$varE,col=4,lwd=2);
   abline(v=fm$burnIn/fm$thin,col=4)

  # lambda (regularization parameter of the Bayesian LASSO)
   lambda<-scan("ETA_MRK_lambda.dat")
   plot(lambda,type="o",col=2,cex=.5,
          xlab="Sample",ylab=expression(lambda),
          main="Regularization parameter");
   abline(h=fm$ETA$MRK$lambda,col=4,lwd=2);
   abline(v=fm$burnIn/fm$thin,col=4)
```

Box S3 shows how to extract estimates, predictions and variance components from the regression model fitted using the script provided in Box 9.

Box S3: Supplementary code for the model fitted in Box 9

```
#1# Predictions
  ## Phenotype prediction
    yHat<-fm$yHat
    tmp<-range(c(y,yHat))
    plot(yHat~y,xlab="Observed",ylab="Predicted",col=2,
            xlim=tmp,ylim=tmp); abline(a=0,b=1,col=4,lwd=2)

#2# Godness of fit and related statistics
   fm$fit
   fm$varE # compare to var(y)

#3# Variance components associated with the genomic and pedigree
  # matrices
  fm$ETA$PED$varU
  fm$ETA$PED$SD.varU

  fm$ETA$MRK$varU
  fm$ETA$MRK$SD.varU

#4# Trace plots
  list.files()
  # Residual variance
  varE<-scan("PGBLUP_varE.dat")
  plot(varE,type="o",col=2,cex=.5);

  #varA and varU
  varA<-scan("PGBLUP_ETA_PED_varU.dat")
  plot(varA,type="o",col=2,cex=.5);

  varU<-scan("PGBLUP_ETA_MRK_varU.dat")
  plot(varU,type="o",col=2,cex=.5)

  plot(varA~varU,col=2,cex=.5,main=paste("Cor= ",round(cor(varU,varA),3),sep=""))

  varG<-varU+varA
  h2<-varG/(varE+varG)
  mean(h2);sd(h2)

  mean(varU/varG)
  mean(varA/varG)
```

Box S4 provides supplementary code for the model fitted in Box 10.

Box S4: Supplementary code for the model fitted in Box 10

```
fm$varE
plot(y~fm$yHat)

plot(scan("RKHS_h=0.25_ETA_K1_varU.dat"),type="o",col=2,cex=0.5)
abline(h=fm$ETA$K1$varU,col=4)
plot(scan("RKHS_h=0.25_varE.dat"),type="o",col=2,cex=0.5)
abline(h=fm$varE,col=4)
```

Box S5 provides supplementary code for the model fitted in Box 11.

Box S5: Supplementary code for the model fitted in Box 11

```
# Posterior mean of the residual variance
fm$varE

# Posterior means of the variances of the kernels
VAR<-c(fm$ETA[[1]]$varU, fm$ETA[[2]]$varU, fm$ETA[[3]]$varU)
names(VAR)<-paste("Variance(h=",h,")",sep="")
barplot(VAR,ylab="Estimated Variance")

# Plots of variance components
varE<-scan("RKHS_KA_varE.dat")
varU1<-scan("RKHS_KA_ETA_1_varU.dat")
varU2<-scan("RKHS_KA_ETA_2_varU.dat")
varU3<-scan("RKHS_KA_ETA_3_varU.dat")
varU<-varU1+varU2+varU3

plot(varE,col=2,type="o",cex=.5,ylab="Residual Variance")
plot(varU,col=2,type="o",cex=.5,ylab="Variance",main="Genomic Variance")
plot(varU1,col=2,type="o",cex=.5,ylab="Variance",main=paste("Variance (h=",h[1],")"))
plot(varU2,col=2,type="o",cex=.5,ylab="Variance",main=paste("Variance (h=",h[2],")"))
plot(varU3,col=2,type="o",cex=.5,ylab="Variance",main=paste("Variance (h=",h[3],")"))
```

Box S6 provides supplementary code for the model fitted in Box 12.

Box S6: Supplementary code for the model fitted in Box 12

```
  # Assesment of correlation in TRN and TST data sets
   cor(fm$yHat[tst],y[tst])    #TST
   cor(fm$yHat[-tst],y[-tst])  #TRN

# Plot of phenotypes versus genomic prediction, by set (TRN/TST)
   plot(y~I(fm$yHat),ylab="Phenotype",
        xlab="Pred. Gen. Value" ,cex=.8,bty="L")
   points(y=y[tst],x=fm$yHat[tst],col=2,cex=.8,pch=19)
   legend("topleft", legend=c("training","testing"),bty="n",
          pch=c(1,19), col=c("black","red"))
   abline(lm(I(y[-tst])~I(fm$yHat[-tst]))$coef,col=1,lwd=2)
   abline(lm(I(y[tst])~I(fm$yHat[tst]))$coef,col=2,lwd=2)
```

Box S7 provides supplementary code for the model fitted in Box 13.

Box S7: Supplementary code for the model fitted in Box 13

```
 # Comparing models using a paired t-test
  colMeans(COR)
  mean(COR[,2]-COR[,1])
  t.test(x=COR[,2],y=COR[,1],paired=TRUE,var.equal=FALSE)

 # Plots of Correlations: Pedigree+Markers vs Pedigree Only
 xy_limits<-range(as.vector(COR))
 plot(COR[,2]~COR[,1],col="red",
      xlim=xy_limits,
      ylim=xy_limits,
      main="E1",
      xlab="Pedigree", ylab="Pedigree+Markers")
 abline(0,1,col="blue")
```

D Regression with Ordinal and Binary Traits

For categorical traits BGLR uses the probit link and the phenotype vector should be coercible to a factor. The type of response is defined by setting the argument `response_type`. By default this argument is set equal to `"Gaussian"`. For binary and ordinal outcomes we should set `response_type="ordinal"`. Box S8 provides a simple example that uses the wheat data set with a discretized phenotype. The second block of code, `#2#`, presents the analysis of a binary outcome, and the third one, `#3#`, that of an ordinal trait. Figure S1 shows, for the binary outcome, a plot of predicted probability (`fmBin$probs`) versus realized value in the TRN and TST datasets.

Box S8: Fitting models with binary and ordinal responses

```
#1# Loading and preparing the input data
 library(BGLR); data(wheat);
 Y<-wheat.Y; X<-wheat.X; A<-wheat.A;
 y<-Y[,1]
 set.seed(123)
 tst<-sample(1:nrow(X),size=150)
 #2# Binary outcome
 yBin<-ifelse(y>0,1,0)
 yBinNA<-yBin; yBinNA[tst]<-NA
 ETA<-list(list(X=X,model="BL"))

 fmBin<-BGLR(y=yBinNA,response_type="ordinal", ETA=ETA,
          nIter=1200,burnIn=200)

 head(fmBin$probs)
 par(mfrow=c(1,2))
 boxplot(fmBin$probs[-tst,2]~yBin[-tst],main="Training",ylab="Estimated prob.")
 boxplot(fmBin$probs[tst,2]~yBin[tst],main="Testing", ylab="Estimated prob.")

#2# Ordinal outcome
 yOrd<-ifelse(y<quantile(y,1/4),1,ifelse(y<quantile(y,3/4),2,3))
 yOrdNA<-yOrd; yOrdNA[tst]<-NA

 ETA<-list(list(X=X,model="BL"))

 fmOrd<-BGLR(y=yOrdNA,response_type="ordinal", ETA=ETA,
          nIter=1200,burnIn=200)
 head(fmOrd$probs)
```

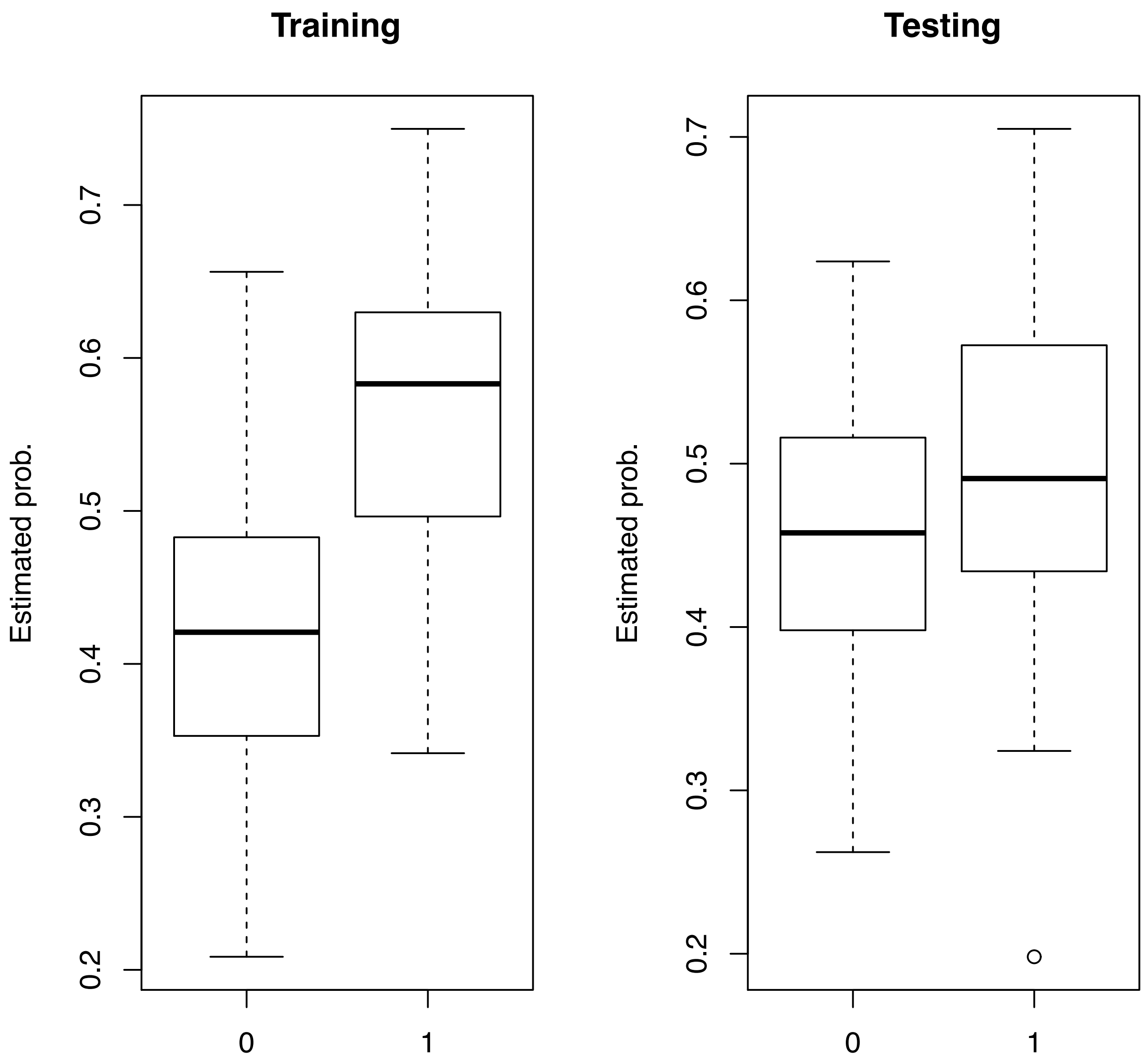

Figure S1: Estimated probability by category, versus observed category (binary response).

E Regression with Censored Outcomes

Box S9 illustrates how to fit a model to a censored trait. Note that in the case of censored trait the response is specified using a triplet (a_i, y_i, b_i) (see Table 2 for further details). For assessment of prediction accuracy (not done in Box S9), one can set $a_i = -\infty$, $y_i = NA$, $b_i = \infty$ for individuals in testing data sets, this way there is no information about the ith phenotype available for the model fit.

Box S9: Fitting censored traits

```
#1# Loading and preparing the input data
 library(BGLR); data(wheat);
 Y<-wheat.Y; X<-wheat.X; A<-wheat.A;
 y<-Y[,1]
 set.seed(123)

#censored
 n<-length(y)
 cen<-sample(1:n,size=200)
 yCen<-y
 yCen[cen]<-NA
 a<-rep(NA,n)
 b<-rep(NA,n)
 a[cen]<-y[cen]-runif(min=0,max=1,n=200)
 b[cen]<-Inf

#models
ETA<-list(list(X=X,model="BL"))

fm<-BGLR(y=yCen,a=a,b=b,ETA=ETA,nIter=12000,burnIn=2000)

cor(y[cen],fm$yHat[cen])
```

File S2

Boxes.R

Available for download at http://www.genetics.org/lookup/suppl/doi:10.1534/genetics.114.164442/-/DC1