SCHOOL OF
PUBLIC HEALTH
Biostatistics Department

The BGLR (Bayesian Generalized Linear Regression) R-Package

By

Gustavo de los Campos & Amit Pataki

(contact: gdeloscampos@gmail.com)

Contents

1. Introduction

The BLR (Bayesian Linear Regression, http://cran.r-project.org/web/packages/BLR/index.html)
package of R (http://cran.r-project.org) implements several types of Bayesian regression models,
including fixed effects, Bayesian Lasso (BL, Park and Casella 2008) and Bayesian Ridge
Regression. BLR can only handle continuous outcomes. Here we introduce BGLR (Bayesian
Generalized Linear Regression). This package extends BLR by allowing regressions for binary and
censored outcomes. Most of the inputs, processes and outputs are as in BLR. Two supporting
documents (BLR_help.pdf and BLR_PerezEtAl.pdf) provide additional information about that.
Here we focus on describing changes in inputs, internal process and outputs introduced to
handle binary and censored outcomes. Users that are not familiar with BLR are strongly
encourage to first read BLR.pdf.

   Censored outcomes.   In BGLR censored outcomes are dealt-with as a missing data
problem. BGLR handles three types of censoring: left, right and interval censored. For an interval
censored data-point the information available is $a_i < y_i < b_i$ where: a_i and b_i are known lower
and upper bounds and y_i is the actual phenotype which for censored data points is un-
observed. Rigth censoring occurs when b_i is also uknown, therefore, the only information
available is $a_i < y_i$. In a time-to-event setting this means that we know that time to event
exceeded the time at censoring given by a_i. Left censoring occurs when a_i is unknown;
therefore, the only information available is: $y_i < b_i$. In BGLR censored outcomes are then
specified with three vectors, $\mathbf{y} = \{y_i\}$, $\mathbf{a} = \{a_i\}$ and $\mathbf{b} = \{b_i\}$. The configuration of the triplet
$\{a_i, y_i, b_i\}$ for un-censored, right-censored, left-censored and interval censored are described in
the table below.

	a	y	b
Un-censored	NA	y_i	NA
Right Censored	a_i	NA	∞
Left Censored	-∞	NA	b_i
Interval Censored	a_i	NA	b_i

Relative to BLR, the only modification introduced in the Gibbs sampler required for handling censored data points consist of sampling, at each iteration of the Gibbs sampler, the censored phenotypes form the corresponding fully-conditional densities which in BGLR are truncated normal densities.

Binary outcomes are modeled using the threshold model, or probit link. Here, probability of success is $p(y_i = 1) = \Phi(\eta_i)$ where $\Phi(.)$ is the standard normal cumulative distribution function (also known as normal probit link) and η_i is a linear predictor which can include the type of fixed or random effects handled by BLR. In order to run a regression for binary outcomes, the response must be coded with 0's (failure) and 1's (success), and the argument family should be set to 'bernoulli' (further details are given in the examples provided below).

2. Structure of the software

The program is provided as a zipped file (BGLR.zip), the folders and files included in BGLR are described in Figure 1.

Files and folders. The package includes four folders: data, which contains a simulated dataset and a wheat dataset (further details are give below), doc, which contains documentation for BGLR and BLR, program which contains the software that implements BGLR and output, this is an empty folder which is used to store outputs from the examples provided in BGLR.pdf.

```
BGLR/
|-- data
|       |-- wheatData.rda
|-- doc
|       |-- BGLR.doc
|       |-- BLR_PerezEtAl.pdf
|       |-- BLR_help.pdf
|-- output
|-- program
        |-- BGLR.rda
        |-- BLR.dll
        |-- BLR.so
```
Figure 1. Folders and Files included in BGLR.

Data. The data folder contains a dataset (**wheatData.rda**) which comprises phenotypic (**Y**, 4 traits), marker (**X**, 1,279 markers) and pedigree (**A**, a matrix containing 2×kinship coefficients derived from pedigree) information for 599 lines of wheat. This dataset is is also available with BLR (type libarary(BLR) ; data(wehat) to access this data via BLR). Further details about this data can be found in Crossa et al. (2010).

2. Running BGLR

In this section we introduce examples that illustrate the use of the BGLR package for regressions using molecular markers and other covariates.

2.1. Loading the BGLR package

Box 1 provides the code required to load BGLR.

Box 1. Loading BGLR
| ```
1    setwd('c:/BGLR/BGLR')
2    library(SuppDists)
3    library(bayesm)
4    libname<-paste('program/BLR',.Platform$dynlib.ext,sep='')
5    dyn.load(libname)
6    load('program/BGLR.rda')
``` |

2.2. Fitting a fixed effects model to a continuous outcome

In the following example we illustrate how fit a 'fixed effects' linear model to a continous outcome using BGLR (line 18 in Box 1). The code in lines 5-9 loads the program. The code in line 10 loads a dataset (`wheatData.rda`) that contains phenotypic and genotypic information of 599 pure lines of wheat, this dataset is also available with the BLR package (de los Campos and Pérez 2010). Lines 15-19 simulates phenotypes using 4 markers. Lines 22-23 define the prior density assigned to the residual variance. Details about the priors used in BLR and on how to choose hyper-parameters are explained in BLR.pdf and in Pérez et al. (2010). The linear model is fitted using BGLR in lines 24-25. The argument y in BGLR is used to provide phenotypes, for continuous outcomes this must be a numeric vector and the argument XF is a design matrix for fixed effects. In addition to phenotypes, we indicate the number of iterations of the Gibbs sampler (6000) and the number that we want to discard as burn-in (1000 in the example). For comparison we include in line 26 code that fits the same linear model via ordinary least squares

using the lm() function. Results from both BGLR and lm are displayed in Figure 1, the code used

to produce this figure is given in lines 27-28 of Box 12.

Box 2. Fitting a fixed effects model to a continuous outcome

```
1    rm(list=ls())
2    setwd('c:/BGLR/BGLR')
3
4   # loads libraries
5    library(SuppDists)
6    library(bayesm)
7    libname<-paste('program/BLR',.Platform$dynlib.ext,sep='')
8    dyn.load(libname)
9    load('program/BGLR.rda')
10   load("data/wheatData.rda")
11
12   setwd("output")
13
14  # simulation of data
15   X<-X[,1:4]
16   N<-nrow(X)
17   b<-c(-2,2,-1,1)
18   error<-rnorm(N)
19   y<-X%*%b+ error
20
21  # fits models
22   DF=5
23   prior=list(varE=list(df=DF,S=var(y)/2*(DF-2)))
24   fm1<-BGLR(y=y,XF=X,nIter=6000,burnIn=1000,
25            prior=prior, saveAt="box2_")
26   fm2<- lm(y~X)
27   plot(fm1$bF~fm2$coeff[-1],pch=19,col=2,cex=1.5,
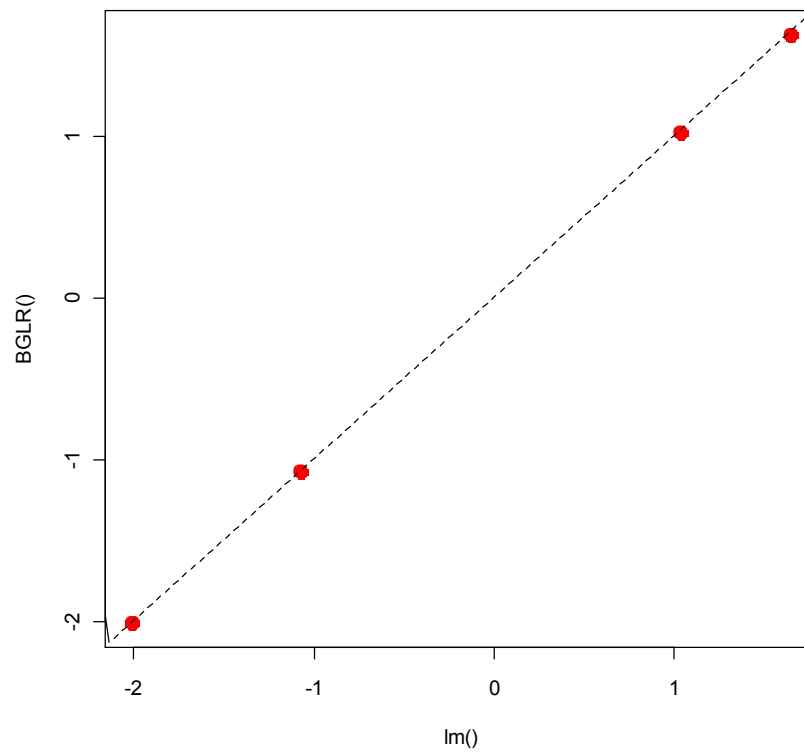28   xlab="lm()", ylab="BGLR()") ; abline(a=0,b=1,lty=2)
```

Figure 1. Estimated effects in a linear model for a continuous outcome (BGLR vs lm).

2.3. Fitting a fixed effects model to a binary outcome

We now turn into an example involving a binary outcome. Using the same simulation used in Box 2, we generate a binary outcome by by dichotomizing the simulated phenotype. The model is fitted using BGLR in lines 21-22. For comparison, we also fit the model using the glm() function of R (line 23). In BGLR we set the argument `family="bernoulli"` (see line 22) to indicate that the response is binary. Note that for binary outcomes we do not have a residual variance parameter, therefore, for this example there is no need to provide a prior. Estimates of effects derived using BGLR and glm are given in Figure 2.

Box 3. Fitting a fixed effects model to a binary outcome

```
1    rm(list=ls())
2    setwd('c:/BGLR/BGLR')
3   # loads libraries
4    load('program/BGLR.rda')
5    load("data/wheatData.rda")
6    setwd("output")
7
8   # simulation of data
9    X<-X[,1:4]
10   N<-nrow(X)
11   b<-c(-2,2,-1,1)
12   error<-rnorm(N)
13   y<-X%*%b+ error
14   yBin<-ifelse(y>0,1,0)
15
16  # fits models
17   fm1<-BGLR(y=yBin,XF=X,nIter=6000,burnIn=1000,
20           family="bernoulli",saveAt="box3_")
21   fm2<- glm(yBin~X,family=binomial(link='probit'))
22
23   plot(fm1$bF~fm2$coeff[-1],pch=19,col=2,cex=1.5,
24   xlab="glm()", ylab="BGLR()") ; abline(a=0,b=1,lty=2)
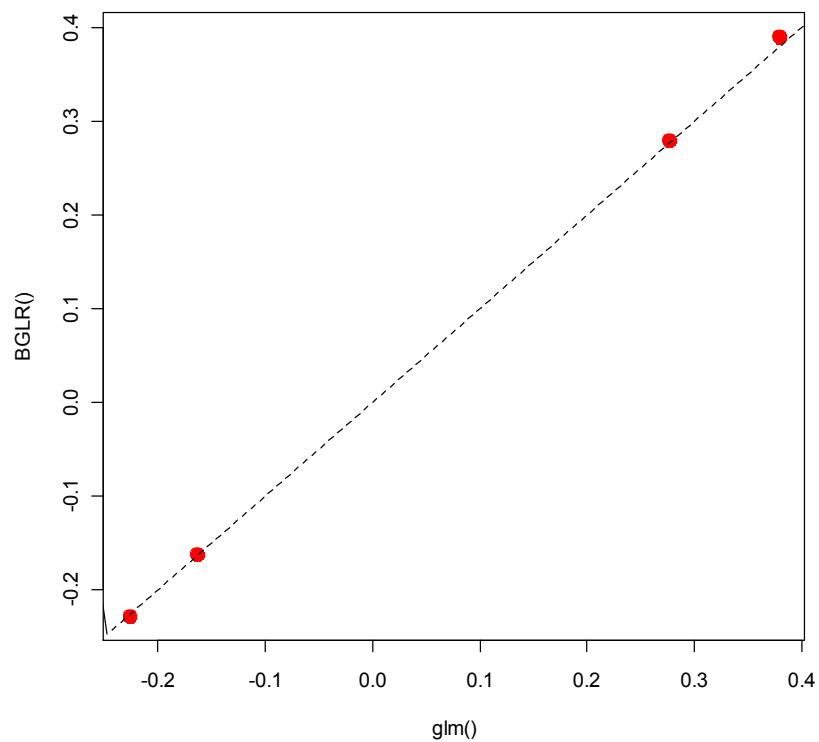25
```

Figure 2. Estimated effects in fixed effects model for a binary outcome (BGLR vs glm)

2.4. Fitting fixed effects model to a right-censored outcome

We now illustrate how to use BGLR to fit a model to a right-censored outcome. The code is given in Box 4. The beginning of the code (lines 1-20) is as in the examples introduced in Box 2 ad 3. In lines 22-26 we generate 200 right-censored data points. These are defined using the conventions explained in Table 1. Subsequently, we fit the model in lines 32-33 to the censored outcome. Relative to un-censored outcomes (see example in Box 2) the only difference here is that the response is specified via 3 vectors (y,a,b) which are defined using the conventions explained in Table 1. For comparison we fit the same model using the surverg function of the survival package (lines 36-41). Figure 3 gives estimates of effects derived from survreg and BGLR.

Figure 3. Estimated effects in fixed effects model for a binary outcome (BGLR vs survreg)

Box 4. Fitting a fixed effects model to a censored outcome

```
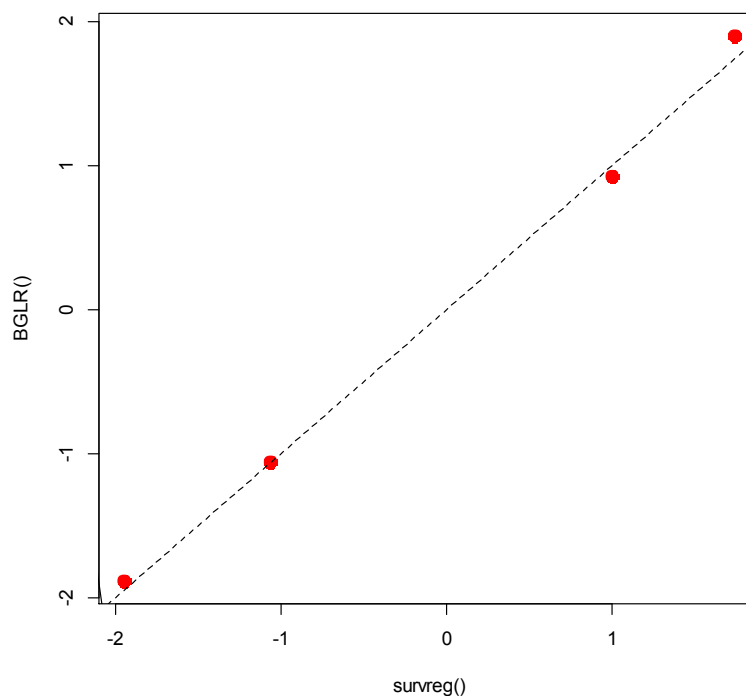1   rm(list=ls())
2  # loading libraries
3   setwd('c:/BGLR/BGLR')
4   library(SuppDists)
5   library(bayesm) ; library(survival)
6   libname<-paste('program/BLR',.Platform$dynlib.ext,sep='')
7   dyn.load(libname)
8   load('program/BGLR.rda')
9   load("data/wheatData.rda")
10   setwd("output")
11
12
13  # simulation of data
14   X<-X[,1:4]
15   N<-nrow(X)
16   b<-c(-2,2,-1,1)
17   error<-rnorm(N)
20   y<-as.vector(X%*%b+ error)
21   n<-nrow(X)
22   cen<-sample(1:n,size=200)
23   yCen<-y
24   yCen[cen]<-NA ; a<-rep(NA,n) ; b<-rep(NA,n)
25   a[cen]<-y[cen]-runif(min=0,max=1,n=200)
26   b[cen]<-Inf
27
28  # fits the model using BGLR
29   DF<-5
30   prior=list(varE=list(df=DF,S=var(y)/2*(DF-2)))
31
32   fm1<-BGLR(y=yCen,a=a,b=b,XF=X,nIter=6000,burnIn=1000,
33            saveAt="box4_")
34
36  # fits the model using survreg
37   event<-ifelse(is.na(yCen),0,1)
38   time<-ifelse(is.na(yCen),a,yCen)
39
40   surv.object<- Surv(time=time,event=event,type='right')
41   fm2<-survreg(surv.object~X, dist="gaussian")
42
43   plot(fm1$bF~fm2$coeff[-1],pch=19,col=2,cex=1.5,
44  xlab="survreg()", ylab="BGLR()") ; abline(a=0,b=1,lty=2)
```

2.5. Fitting marker effects as random

We now turn into the problem of using BGLR for fitting a Whole-Genome Regression (WGR) model to either continuous, binary or censored outcomes. In these models, the number of predictors typically exceeds the number of phenotypes; therefore, shrinkage estimation procedures are commonly used. BGLR offers two alternative shrinkage (Bayesian) estimation methods: Bayesian Ridge Regression (BRR) and the Bayesian Lasso (BL, Park and Casella 2008). Here we illustrate how to fit models for continuous, binary and a censored outcome using the BL. For the BL we need to provide a prior to the regularization parameter () which controls the extent of shrinkage of estimates of effects. A discussion of how to choose these hyper-parameters based on prior information about tait heritability and on the number of markers involved is given in BLR.pdf and in Pérez et al. (2010).

In the example given in Box 5 we fit the BL using, for the 599 wheat lines available in the wheat dataset, 1,279 markers. Lines 3-8 give the code required for loading BGLR, line 9 loads the wheat dataset. In line 15 we extract one of the four phenotypes, this will be used as a continuous response. Subsequently we generate (lines 20-24) a right-censored outcome by censoring 200 out of the 599 records. These lines prepare the triplets (y,a,b) needed to specify the censored outcome in BGLR. Finally, in line 27 we generate a binary outcome. Lines 35-40 are used to fit the models. As the number of markers included in the model increases the number of iterations required for convergence also increases, in the example of Box 5, and only for illustration purposes, we use 12,000 iterations; however, convergence with large-p may require running much longer chains.

Box 6 gives code that illustrates how to extract estimates of marker effects and predictions from the fitted model.

Box 5. Fitting a Whole Genome Regression Using the Bayesian LASSO for continuous, censored and binary outcomes

```
1    rm(list=ls())
2   # loading libraries
3    setwd('c:/BGLR/BGLR')
4    library(SuppDists)
5    library(bayesm)
6    libname<-paste('program/BLR',.Platform$dynlib.ext,sep='')
7    dyn.load(libname)
8    load('program/BGLR.rda')
9    load("data/wheatData.rda")
10   setwd("output")
11   n<- nrow(Y)
12
13  # extracts phenotypes
14   #continous
15   y<-Y[,1]
16
17   #censored
20   cen<-sample(1:n,size=200)
21   yCen<-y
22   yCen[cen]<-NA ; a<-rep(NA,n) ; b<-rep(NA,n)
23   a[cen]<-y[cen]-runif(min=0,max=1,n=200)
24   b[cen]<-Inf
25
26   #binary
27   yBin<-ifelse(y>0,1,0)
28
29  # prior
30   DF<-5
31   prior=list(varE=list(df=DF,S=var(y)/2*(DF-2)),
32        lambda=list(value=25,type='random', rate=1e-4,shape=0.55)
33             )
34  # models
35  fm1<-BGLR(y=y,a=a,b=b,XL=X,nIter=12000,burnIn=2000,
36           saveAt="box5_con_",prior=prior)
37  fm2<-BGLR(y=yCen,a=a,b=b,XL=X,nIter=12000,burnIn=2000,
38           saveAt="box5_cen_",prior=prior)
39  fm3<-BGLR(y=yBin,a=a,b=b,XL=X,nIter=12000,burnIn=2000,
40           saveAt="box5_bin_",prior=prior, family='bernoulli')
```

2.6. Extracting estimates of marker effects and predictions

Box 6 illustrates how to extract: the estimated posterior means and posterior standard deviations of marker effects (e.g., `fm$bL` and `fm$SD.bL`, see lines 3 and 8, respectively) and posterior means of the linear predictor (e.g., `fm$yHat`). For binary and censored outcomes the linear posterior mean of the linear predictor constitutes an estimate of the conditional expectation. For binary outcomes, BGLR uses the probit link; therefore an estimate of the expected value of the response, or probability of success, can be obtained by evaluating the standard normal cumulative distribution function at the posterior mean of the linear predictor (see line 20 in Box 6).

Box 6. Extracting and Displaying Estimates of Marker Effects and Predictions

```
1
2    ## Vulcano plot (posterior SD vs estimated effects)
3    plot(fm1$SD.bL~fm1$bL,col=2,
4         main='Vulcano Plot (continuous outcome)',
5         xlab='Estimated Effect',ylab='Est. Posterior SD')
6
7    ## Estimated effects, continuous versus censored
8    plot(fm1$bL~fm2$bL,col=2, main='Estimated Effects',
9         xlab='Censored', ylab='Continuos')
10
11   ## Predictions: continuous versus censored outcome
12   plot(fm1$yHat~fm2$yHat,col=2, main='Predictions',
13        xlab='Censored', ylab='Continuos')
14
15   ## Estimated effects, continuous versus binary
16   plot(fm1$ bL~fm3$bL,col=2, main='Estimated Effects',
17        xlab='Binary', ylab='Continuos')
18
19   ## Predictions: continuous versus binary outcome
20   plot(fm1$yHat~pnorm(fm3$yHat),col=2, main='Predictions',
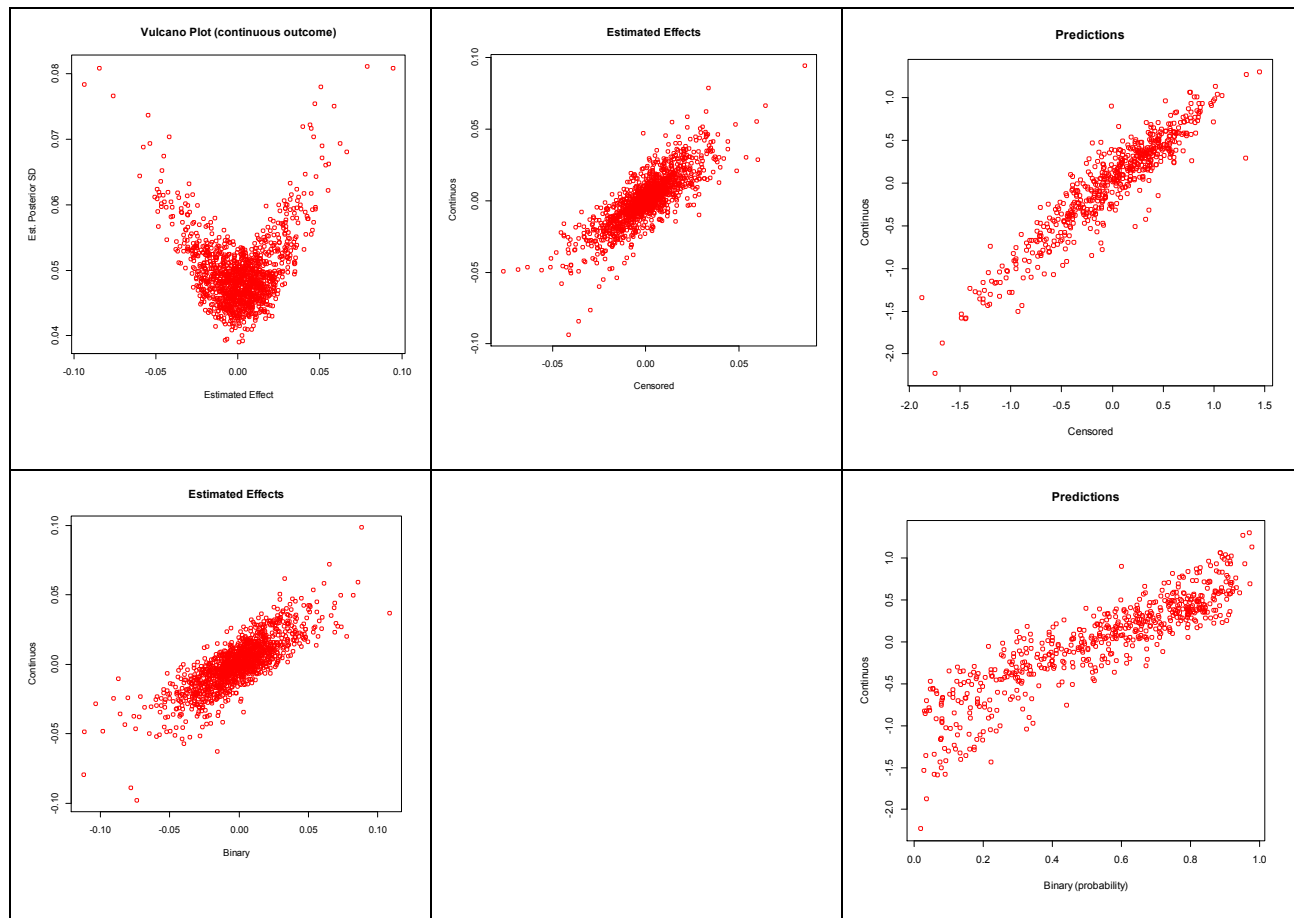21        xlab='Binary (probability)', ylab='Continuos')
```

Figure 4. Estimates of marker effects and predictions obtained with the BL fitted to a continuous, censored and binary outcome.

2.7. Predicting un-observed outcomes using BGLR

We close this note by illustrating how to use BGLR for the prediction of yet-to-be observed phenotypes. In principle there are at least two ways of carrying out this task. One possibility is to partition the data (both predictors and response) into training and a validation dataset, the training dataset is provided to BGLR to derive parameter estimates which could then be used to predict observations in the validating dataset. An alternative is to provide the whole data to BGLR with the response values of the observations in the validation set replaced with missing values. BGLR will return predictions for these data-points as well and such

15

predictions can be used to assess the ability of the model to predict un-observed phenotypes. In the case of continuous and binary outcomes this is done simply by setting the entries of y corresponding to the validation dataset equal to NA (see example below); for censored outcomes, the triplets corresponding to the validation set needs to be set to ($a_i=-\infty$, y_i=NA, $b_i=\infty$) so that these are completely un-informative.

Prediction of binary outcomes. The example in Box 7 illustrates how to derive predictions for a validation dataset in case of a binary outcome. Lines 1-20 are used to load libraries and the dataset and to define the prior. These are essentially as in our previous examples. In lines 22-23 we generate a validation set by setting 100 randomly chosen entries of the response to missing values. The model is fitted in lines 25-26. Lines 28-29 illustrate how to calculate mean-squared prediction error and 'area under the curve'.

Box 7. Fitting a Whole Genome Regression Using the Bayesian LASSO for continuous, censored and binary outcomes

```
1    rm(list=ls())
2   # loading libraries
3    setwd('c:/BGLR/BGLR')
4    library(SuppDists) ; library(bayesm); library(pROC)
5    libname<-paste('program/BLR',.Platform$dynlib.ext,sep='')
6    dyn.load(libname) ;  load('program/BGLR.rda')
7    load("data/wheatData.rda")
8    setwd("output")
9
10  # extracts phenotypes
11   #continous
12    y<-Y[,1]
13   #binary
14    yBin<-ifelse(y>0,1,0)
15   # prior
16    prior=list(lambda=list(value=25,type='random',
17                           rate=1e-4,shape=0.55))
20  # generates testing dataset
21    tst<-sample(1:599,size=100,replace=FALSE)
22    yNA<-yBin ; yNA[tst]<-NA
23
24  # model
25    fm<-BGLR(y=yNA, XL=X,nIter=12000,burnIn=2000,
26          saveAt="box7_",prior=prior,family="bernoulli")
27
28  mean((yBin[tst]-pnorm(fm$yHat[tst]))^2) # mean-sq. error
29  auc(response=yBin[tst],predictor=fm$yHat[tst])
```

Prediction of censored outcomes. The example in Box 8 illustrates how to derive predictions for a validation dataset in case of a censored outcome. Lines 1-26 are used to load libraries and the dataset and to define the prior. These are essentially as in our previous examples. In lines 25-28 we generate a validation using 100 lines randomly chosen among the un-censored observations. Note that in order for these phenotypes to be un-informative we need to set the triplets of the lines in the validation dataset to $(a_i=-\infty, y_i=NA, b_i=\infty)$. The model is fitted in lines 28-29 and prediction accuracy is quantified in lines .

Box 7. Fitting a Whole Genome Regression Using the Bayesian LASSO for continuous, censored and binary outcomes

```
1    rm(list=ls())
2   # loading libraries
3    setwd('BGLR')
4    library(SuppDists) ; library(bayesm); library(pROC)
5    libname<-paste('program/BLR',.Platform$dynlib.ext,sep='')
6    dyn.load(libname) ;  load('program/BGLR.rda')
7    load("data/wheatData.rda") ; setwd("output")
8
9   # extracts phenotypes
10   #censored
11    cen<-sample(1:n,size=200)
12    y<-Y[,1] ; yCen<-y ; n=length(y)
13    yCen[cen]<-NA ; a<-rep(NA,n) ; b<-rep(NA,n)
14    a[cen]<-y[cen]-runif(min=0,max=1,n=200)
15    b[cen]<-Inf
16
17  # prior
20    DF<-5
21    prior=list(varE=list(df=DF,S=0.5*(DF-2)),
22              lambda=list(value=25,type='random',
23                         rate=1e-4,shape=0.55))
24  # generates testing dataset
25    tst<-sample(which(!is.na(yCen)),size=100,replace=FALSE)
26    yNA<-yCen ; yNA[tst]<-NA
27    aNA<-a   ; aNA[tst]<- -Inf
28    bNA<-b   ; bNA[tst]<- Inf
29
30  # model
31    fm<-BGLR(y=yNA,a=aNA,b=bNA,XL=X,nIter=12000,burnIn=2000,
32            saveAt="box8_",prior=prior)
33    cor(fm$yHat[tst],yCen[tst])
```

References

de los Campos, G., and P. Pérez. 2010. *BLR: Bayesian Linear Regression. R Package Version 1.2*. http://cran.r-project.org/web/packages/BLR/index.html.

Crossa, J., G. de los Campos, P. Perez, D. Gianola, J. Burgueño, J. L Araus, D. Makumbi, et al. 2010. "Prediction of Genetic Values of Quantitative Traits in Plant Breeding Using Pedigree and Molecular Markers." *Genetics* 186 (2): 713–724.

Park, T., and G. Casella. 2008. "The Bayesian Lasso." *Journal of the American Statistical Association* 103 (482): 681–686.

Pérez, Paulino, Gustavo de los Campos, José Crossa, and Daniel Gianola. 2010. "Genomic-Enabled Prediction Based on Molecular Markers and Pedigree Using the Bayesian Linear Regression Package in R." *The Plant Genome Journal* 3 (2): 106–116. doi:10.3835/plantgenome2010.04.0005.