The BGLR (Bayesian Generalized Linear Regression) R-Package

By

Gustavo de los Campos, Amit Pataki & Paulino Pérez

(August-2013)

(contact: gdeloscampos@gmail.com)

Contents

1. Introduction 2

2. Structure of the software 3

3. Running BGLR 4

3.1. Loading the BGLR package 4
3.2. Fitting a fixed effects model to a continuous outcome 4
3.3. Fitting a fixed effects model to a binary outcome 6
3.4. Fitting fixed effects model to a right-censored outcome 8
3.5. Fitting marker effects as random 10
3.6. Extracting estimates of marker effects and predictions 12
3.7. Predicting un-observed outcomes using BGLR 13

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. We have produced a modified (beta) version of BLR (BGLR=Bayesian Generalized Linear Regression) that extends BLR by allowing regressions for binary and censored outcomes. Most of the inputs, processes and outputs are as in BLR. 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 encouraged to first read the BLR user's manual and Pérez et al. (2010). Future developments will be released first in the R-forge webpage https://r-forge.r-project.org/projects/bglr/ and subsequently as R-packages.

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. Right censoring occurs when b_i is also unknown, 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 $\Phi(\cdot)$ is the standard normal cumulative distribution function (also known as normal probit link) and η_i is a linear predictor, which can include fixed or random effects, handled by BGLR. 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 `response_type` should be set to `'ordinal'` (further details are given in the examples provided below).

2. Structure of the software

The program is provided as an R package that can be downloaded from http://r-forge.r-project.org/R/?group_id=1525. The package includes several datasets. Here we describe the wheat dataset that have been used in several publications.

*The **wheat** dataset* 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. The data can be loaded within R typing `library(BGLR)` and then `data(wheat)`. Further details about this data can be found in Crossa et al. (2010).

3. Running BGLR

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

3.1. Loading the BGLR package

Box 1 provides the code required to load BGLR.

Box 1. Loading BGLR

```
1  setwd(tempdir()) #Set working directory
2  library(BGLR)
```

3.2. Fitting a fixed effects model to a continuous outcome

In the following example we illustrate how fit a 'fixed effects' linear model to a continuous outcome using BGLR (line 21 in Box 2). The code in lines 5-7 loads the program and the wheat dataset 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).

Phenotypes are simulated in lines 10-14. The prior assigned to the residual variance is defined in lines 17-18 Details about the priors used in BGLR and on how to choose hyper-parameters are explained in Pérez et al. (2010). The linear model is fitted using BGLR in lines 19-21. The argument `y` in BGLR is used to provide phenotypes, for continuous outcomes this must be a numeric vector and a list with predictors whose effects will be considered as fixed. 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 24 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 2.

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

```
1  rm(list=ls())
2  setwd(tempdir())
3
4  #loads BGLR & Data
5  library(BGLR)
6  data(wheat)
7  X<-wheat.X
8
9  #simulation of data
10 X<-X[,1:4]
11 N<-nrow(X)
12 b<-c(-2,2,-1,1)
13 error<-rnorm(N)
14 y<-as.vector(X%*%b+ error)
15
16 #fits model using BGLR
17 DF<-5
18 S<-var(y)/2*(DF-2)
19 ETA<-list(list(X=X,model='FIXED'))
20
21 fm1<-BGLR(y=y,ETA=ETA,nIter=6000,burnIn=1000,df0=DF,S0=S)
22
23 #fits the same model using lm()
24 fm2<- lm(y~X)
25
26 #compares results from BGLR() & lm()
27 plot(fm1$ETA[[1]]$b~fm2$coeff[-1],pch=19,col=2,cex=1.5,
28 xlab="lm()", ylab="BGLR()"); abline(a=0,b=1,lty=2)
```

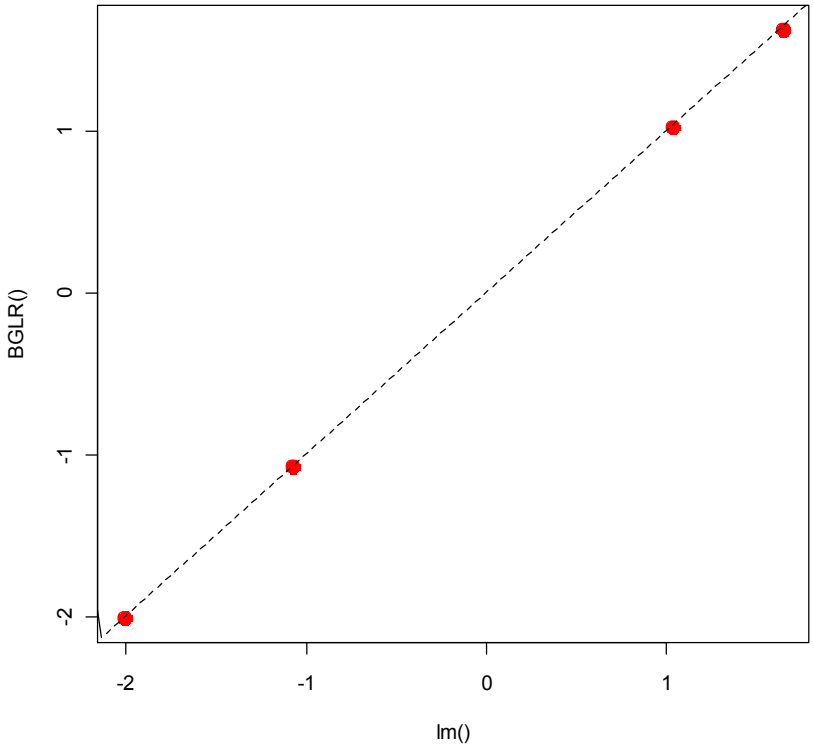

Figure 1. Estimated effects in a linear model for a continuous outcome (BGLR vs lm).

3.3. 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 dichotomizing the simulated phenotype (see line 20 of Box 3). The model is fitted using BGLR in lines 23-25. For comparison, we also fit the model using the glm() function of R (line 27). In BGLR we set the argument `response_type="ordinal"` (see line 24) 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

```
rm(list=ls())
setwd(tempdir())

#loads BGLR & Data
library(BGLR)
data(wheat)
X=wheat.X

#simulation of data
X<-X[,1:4]
N<-nrow(X)
b<-c(-2,2,-1,1)
error<-rnorm(N)
y<-as.vector(X%*%b+ error)
yBin<-ifelse(y>0,1,0)

#fits models
ETA<-list(list(X=X,model='FIXED'))
fm1<-BGLR(y=yBin,response_type='ordinal',ETA=ETA,
          nIter=6000,burnIn=1000)

fm2<- glm(yBin~X,family=binomial(link='probit'))

plot(fm1$ETA[[1]]$b~fm2$coeff[-1],pch=19,col=2,cex=1.5,
xlab="glm()", ylab="BGLR()") ; abline(a=0,b=1,lty=2)
```

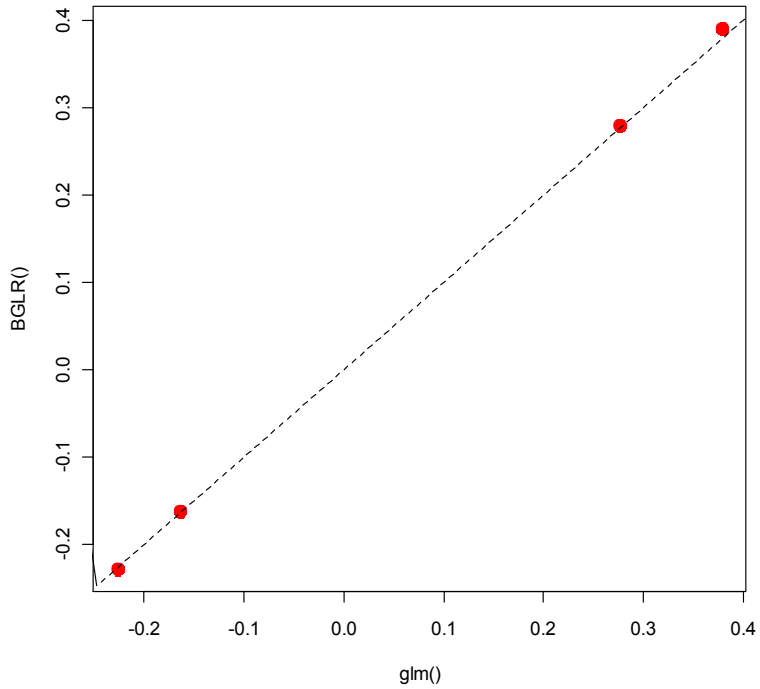

Figure 2. Estimated effects in fixed effects model for a binary outcome (BGLR vs glm)

3.4. 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 1-17) is as in the examples introduced in Box 2 ad 3. In lines 18-24 we generate 200 right-censored data points. These are defined using the conventions explained in Table 1. Subsequently, we fit the model using `BGLR()` in line 30. 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 `survreg()` function of the survival package (lines 34-38). Figure 3 gives estimates of effects derived from `survreg()` and `BGLR()`.

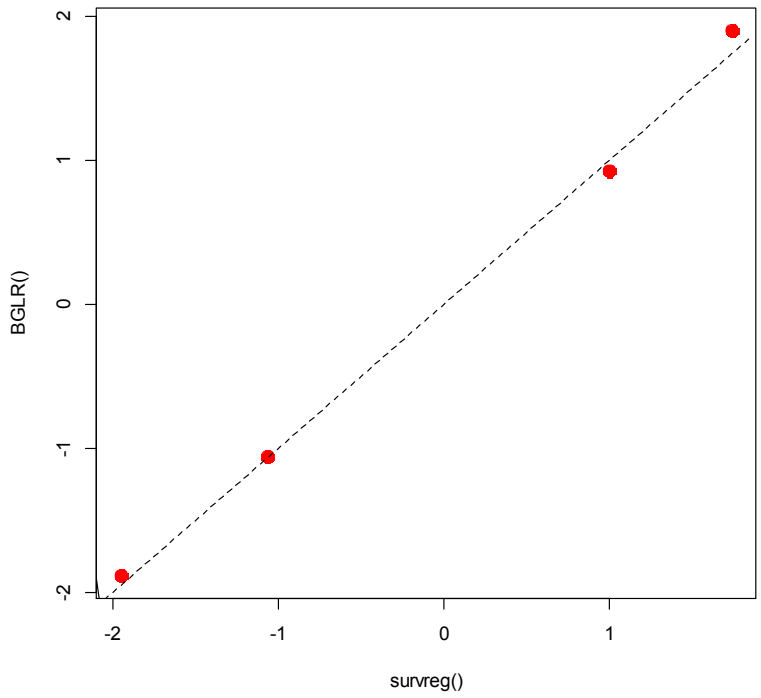

Figure 3. Estimated effects in fixed effects model for a binary outcome (BGLR vs survreg)

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

```
rm(list=ls())
setwd(tempdir())

#loading libraries
library(BGLR)
library(survival)

#loading data included in BGLR
data(wheat)

#simulation of data
X<-wheat.X[,1:4]
N<-nrow(X)
b<-c(-2,2,-1,1)
error<-rnorm(N)
y<-as.vector(X%*%b+ error)

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

DF<-5
S<-var(y)/2*(DF-2)
ETA<-list(list(X=X,model='FIXED'))

fm1<-BGLR(y=y,a=a,b=b,ETA=ETA,nIter=6000,burnIn=1000,
          df0=DF,S0=S)

#fits the model using survreg
event<-ifelse(is.na(yCen),0,1)
time<-ifelse(is.na(yCen),a,yCen)

surv.object<-Surv(time=time,event=event,type='right')
fm2<-survreg(surv.object~X, dist="gaussian")

plot(fm1$ETA[[1]]$b~fm2$coeff[-1],pch=19,col=2,cex=1.5,
     xlab="survreg()", ylab="BGLR()")
abline(a=0,b=1,lty=2)
```

3.5. Fitting marker effects as random

We now turn into the problem of using BGLR for fitting a Whole-Genome Regression (WGR) model to 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 several shrinkage (Bayesian) estimation methods, for example: 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 (λ) which controls the extent of shrinkage of estimates of effects. A discussion of how to choose these hyper-parameters based on prior information about trait heritability and on the number of markers involved is given 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 4-7 give the code required for loading BGLR and the wheat dataset. In line 11 we extract one of the four phenotypes, this will be used as a continuous response. In line 14 we extract the genotypes. Subsequently we generate (lines 16-23) 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 26 we generate a binary outcome. Lines 28-42 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

```
rm(list=ls())
setwd(tempdir())

#loading libraries
library(BGLR)
library(survival)
data(wheat)

#extracts phenotypes
#continous
y<-wheat.Y[,1]

#Extract genotypes
X<-wheat.X

n<- length(y)

#censored
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

#binary
yBin<-ifelse(y>0,1,0)

#prior
DF<-5
S<-var(y)/2*(DF-2)

#models
ETA<-list(list(X=X,model='BL',lambda=25,type='gamma',
               rate=1e-4,shape=0.55))
fm1<-BGLR(y=y,ETA=ETA,nIter=12000,burnIn=2000,
          df0=DF,S0=S)

fm2<-BGLR(y=yCen,a=a,b=b,ETA=ETA,nIter=12000,burnIn=2000,
          df0=DF,S0=S)

fm3<-BGLR(y=yBin,response_type='ordinal', ETA=ETA,
          nIter=12000,burnIn=2000)

```

3.6. Extracting estimates of marker effects and predictions

Box 6 illustrates how to extract: the estimated posterior means and posterior standard deviations of marker effects (see lines 3-8) and posterior means of the linear predictor (e.g., `fm$yHat`, see line 13). 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 22 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$ETA[[1]]$b~fm1$ETA[[1]]$SD.b,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$ETA[[1]]$b~fm2$ETA[[1]]$b,col=2,
9        main='Estimated Effects',
10       xlab='Censored', ylab='Continuos')
11
12  ##Predictions: continuous versus censored outcome
13  plot(fm1$yHat~fm2$yHat,col=2, main='Predictions',
14       xlab='Censored', ylab='Continuos')
15
16  ##Estimated effects, continuous versus binary
17  plot(fm1$ETA[[1]]$b~fm3$ETA[[1]]$b,col=2,
18       main='Estimated Effects',
19       xlab='Binary', ylab='Continuos')
20
21  ##Predictions: continuous versus binary outcome
22  plot(fm1$yHat~pnorm(fm3$yHat),col=2, main='Predictions',
23       xlab='Binary (probability)', ylab='Continuos')
```

3.7. 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 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. The code in lines 1-19 loads libraries and the wheat dataset and defines the prior density and sets predictors. These lines are essentially as in our previous examples. In lines 21-24 we generate a validation set by setting 100 randomly chosen entries of the response to missing values. The model is fitted in lines 26-27. Lines 29-30 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   setwd(tempdir())
3
4   #loading libraries
5   library(BGLR)
6   library(pROC)
7   data(wheat)
8
9   #extracts phenotypes
10  #continous
11
12  y<-wheat.Y[,1]
13  X<-wheat.X
14
15  #binary
16  yBin<-ifelse(y>0,1,0)
17
18  ETA<-list(list(X=X,model='BL',lambda=25,type='gamma',
19                 rate=1e-4,shape=0.55))
20
21  #generates testing dataset
22  tst<-sample(1:599,size=100,replace=FALSE)
23  yNA<-yBin
24  yNA[tst]<-NA
25
26  fm<-BGLR(y=yNA,response_type='ordinal', ETA=ETA,
27           nIter=12000,burnIn=2000)
28
29  mean((yBin[tst]-pnorm(fm$yHat[tst]))^2) # mean-sq. error
30  auc(response=yBin[tst],predictor=fm$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 1-24 are used to load libraries and the dataset and to define the prior. These are essentially as in our previous examples. In lines 32-36 we generate a validation set 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 39-40 and prediction accuracy is quantified in line 41.

Box 8. Fitting a Whole Genome Regression Using the Bayesian LASSO for continuous, censored and binary outcomes

```
rm(list=ls())
setwd(tempdir())

#loading libraries
library(BGLR)
library(survival)
data(wheat)

#extracts phenotypes
#continous
y<-wheat.Y[,1]

#Extract genotypes
X<-wheat.X

n<- length(y)

#censored
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

#Set prior and predictors
DF<-5
S<-var(y)/2*(DF-2)

ETA<-list(list(X=X,model='BL',lambda=25,type='gamma',
               rate=1e-4,shape=0.55))

#generates testing dataset
tst<-sample(which(!is.na(yCen)),size=100,replace=FALSE)
yNA<-yCen ; yNA[tst]<-NA
aNA<-a   ; aNA[tst]<- -Inf
bNA<-b   ; bNA[tst]<- Inf

#model
fm<-BGLR(y=yCen,a=a,b=b,ETA=ETA,nIter=12000,burnIn=2000,
          df0=DF,S0=S)
cor(fm$yHat[tst],yCen[tst])
```

Acknowledgments. Financial support from NIH P30 Administrative supplement (UAB-Nutrition Obesity Research Center) and NIH grants R01GM101219-01 and R01GM099992-01A1 are gratefully acknowledged.

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.