

Package ‘BGLR’

November 11, 2018

Version 1.0.8

Date 2018-11-24

Title Bayesian Generalized Linear Regression

Author Gustavo de los Campos, Paulino Perez Rodriguez,

Maintainer Paulino Perez Rodriguez <perpdgo@colpos.mx>

Depends R (>= 3.5.0)

Imports truncnorm

Description Bayesian Generalized Linear Regression.

LazyLoad true

License GPL-3

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-11-11 07:20:03 UTC

R topics documented:

BGLR	2
BLR	5
getVariances	11
mice	12
mice.A	12
mice.pheno	13
mice.X	13
plot.BGLR	13
predict.BGLR	14
readBinMat	15
read_bed	16
read_ped	17
residuals.BGLR	18
summary.BGLR	19
wheat	20
wheat.A	21

wheat.sets	21
wheat.X	22
wheat.Y	22
write_bed	22

Index	24
--------------	-----------

BGLR	<i>Bayesian Generalized Linear Regression</i>
------	---

Description

The BGLR ('Bayesian Generalized Linear Regression') function fits various types of parametric and semi-parametric Bayesian regressions to continuous (censored or not), binary and ordinal outcomes.

Usage

```
BGLR(y, response_type = "gaussian", a=NULL, b=NULL,ETA = NULL, nIter = 1500,
      burnIn = 500, thin = 5, saveAt = "", S0 = NULL,
      df0 =5, R2 = 0.5, weights = NULL,
      verbose = TRUE, rmExistingFiles = TRUE, groups=NULL)
```

Arguments

y	(numeric, <i>n</i>) the data-vector (NAs allowed).
response_type	(string) admits values "gaussian" or "ordinal". The Gaussian outcome may be censored or not (see below). If response_type="gaussian", y should be coercible to numeric. If response_type="ordinal", y should be coercible to character, and the order of the outcomes is determined based on the alphanumeric order ($0 < 1 < 2 \dots < a < b \dots$). For ordinal traits the probit link is used.
a, b	(numeric, <i>n</i>) only required for censored outcomes, a and b are vectors specifying lower and upper bounds for censored observations, respectively. The default value, for non-censored and ordinal outcomes, is NULL (see details).
ETA	(list) This is a two-level list used to specify the regression function (or linear predictor). By default the linear predictor (the conditional expectation function in case of Gaussian outcomes) includes only an intercept. Regression on covariates and other types of random effects are specified in this two-level list. For instance: ETA=list(list(X=W, model="FIXED"), list(X=Z,model="BL"), list(K=G,model="RKHS")), specifies that the linear predictor should include: an intercept (included by default) plus a linear regression on W with regression coefficients treated as fixed effects (i.e., flat prior), plus regression on Z, with regression coefficients modeled as in the Bayesian Lasso of Park and Casella (2008) plus and a random effect with co-variance structure G. For linear regressions the following options are implemented: FIXED (Flat prior), BRR (Gaussian prior), BayesA (scaled-t prior), BL (Double-Exponential

prior), BayesB (two component mixture prior with a point of mass at zero and a scaled-t slab), BayesC (two component mixture prior with a point of mass at zero and a Gaussian slab). In linear regressions X can be the incidence matrix for effects or a formula (e.g. `~factor(sex) + age`) in which case the incidence matrix is created internally using the `model.matrix` function of R. For Gaussian processes (RKHS) a co-variance matrix (K) must be provided. Further details about the models implemented in BGLR see the vignettes in the package or <http://genomics.cimmyt.org/BGLR-extdoc.pdf>.

weights	(numeric, <i>n</i>) a vector of weights, may be NULL. If weights is not NULL, the residual variance of each data-point is set to be proportional to the inverse of the squared-weight. Only used with Gaussian outcomes.
nIter, burnIn, thin	(integer) the number of iterations, burn-in and thinning.
saveAt	(string) this may include a path and a pre-fix that will be added to the name of the files that are saved as the program runs.
S0, df0	(numeric) The scale parameter for the scaled inverse-chi squared prior assigned to the residual variance, only used with Gaussian outcomes. In the parameterization of the scaled-inverse chi square in BGLR the expected values is $S0/(df0-2)$. The default value for the df parameter is 5. If the scale is not specified a value is calculated so that the prior mode of the residual variance equals $var(y)*R2$ (see below). For further details see the vignettes in the package or http://genomics.cimmyt.org/BGLR-extdoc.pdf .
R2	(numeric, $0 < R2 < 1$) The proportion of variance that one expects, a priori, to be explained by the regression. Only used if the hyper-parameters are not specified; if that is the case, internally, hyper-parameters are set so that the prior modes are consistent with the variance partition specified by R2 and the prior distribution is relatively flat at the mode. For further details see the vignettes in the package or http://genomics.cimmyt.org/BGLR-extdoc.pdf .
verbose	(logical) if TRUE the iteration history is printed, default TRUE.
rmExistingFiles	(logical) if TRUE removes existing output files from previous runs, default TRUE.
groups	(factor) a vector of the same length of y that associates observations with groups, each group will have an associated variance component for the error term.

Details

BGLR implements a Gibbs sampler for a Bayesian regression model. The linear predictor (or regression function) includes an intercept (introduced by default) plus a number of user-specified regression components (X) and random effects (u), that is:

$$\eta = 1\mu + X_1\beta_1 + \dots + X_p\beta_p + u_1 + \dots + u_q$$

The components of the linear predictor are specified in the argument ETA (see above). The user can specify as many linear terms as desired, and for each component the user can choose the prior density to be assigned. The distribution of the response is modeled as a function of the linear predictor.

For Gaussian outcomes, the linear predictor is the conditional expectation, and censoring is allowed. For censored data points the actual response value (y_i) is missing, and the entries of the vectors a and b (see above) give the lower and upper bound for y_i . The following table shows the configuration of the triplet (y, a, b) for un-censored, right-censored, left-censored and interval censored.

	a	y	b
Un-censored	NULL	y_i	NULL
Right censored	a_i	NA	∞
Left censored	$-\infty$	NA	b_i
Interval censored	a_i	NA	b_i

Internally, censoring is dealt with as a missing data problem.

Ordinal outcomes are modelled using the probit link, implemented via data augmentation. In this case the linear predictor becomes the mean of the underlying liability variable which is normal with mean equal to the linear predictor and variance equal to one. In case of only two classes (binary outcome) the threshold is set equal to zero, for more than two classes thresholds are estimated from the data. Further details about this approach can be found in Albert and Chib (1993).

Value

A list with estimated posterior means, estimated posterior standard deviations, and the parameters used to fit the model. See the vignettes in the package (or <http://genomics.cimmyt.org/BGLR-extdoc.pdf>) for further details.

Author(s)

Gustavo de los Campos, Paulino Perez Rodriguez,

References

- Albert J., S. Chib. 1993. Bayesian Analysis of Binary and Polychotomus Response Data. *JASA*, **88**: 669-679.
- de los Campos G., H. Naya, D. Gianola, J. Crossa, A. Legarra, E. Manfredi, K. Weigel and J. Cotes. 2009. Predicting Quantitative Traits with Regression Models for Dense Molecular Markers and Pedigree. *Genetics* **182**: 375-385.
- de los Campos, G., D. Gianola, G. J. M., Rosa, K. A., Weigel, and J. Crossa. 2010. Semi-parametric genomic-enabled prediction of genetic values using reproducing kernel Hilbert spaces methods. *Genetics Research*, **92**:295-308.
- Park T. and G. Casella. 2008. The Bayesian LASSO. *Journal of the American Statistical Association* **103**: 681-686.
- Spiegelhalter, D.J., N.G. Best, B.P. Carlin and A. van der Linde. 2002. Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* **64** (4): 583-639.

Examples

```
## Not run:
#Demos
library(BGLR)

#BayesA
demo(BA)

#BayesB
demo(BB)

#Bayesian LASSO
demo(BL)

#Bayesian Ridge Regression
demo(BRR)

#BayesCpi
demo(BayesCpi)

#RKHS
demo(RKHS)

#Binary traits
demo(Bernoulli)

#Ordinal traits
demo(ordinal)

#Censored traits
demo(censored)

## End(Not run)
```

BLR

Bayesian Linear Regression

Description

The BLR ('Bayesian Linear Regression') function was designed to fit parametric regression models using different types of shrinkage methods. An earlier version of this program was presented in de los Campos *et al.* (2009).

Usage

```
BLR(y, XF, XR, XL, GF, prior, nIter, burnIn, thin,thin2,saveAt,
    minAbsBeta,weights)
```

Arguments

<code>y</code>	(numeric, n) the data-vector (NAs allowed).
<code>XF</code>	(numeric, $n \times pF$) incidence matrix for β_F , may be NULL.
<code>XR</code>	(numeric, $n \times pR$) incidence matrix for β_R , may be NULL.
<code>XL</code>	(numeric, $n \times pL$) incidence matrix for β_L , may be NULL.
<code>GF</code>	(list) providing an <code>\$ID</code> (integer, n) linking observations to groups (e.g., lines or sires) and a (co)variance structure (<code>\$A</code> , numeric, $pU \times pU$) between effects of the grouping factor (e.g., line or sire effects). Note: <code>ID</code> must be an integer taking values from 1 to pU ; <code>ID[i]=q</code> indicates that the i th observation in <code>y</code> belongs to cluster q whose (co)variance function is in the q th row (column) of <code>A</code> . <code>GF</code> may be NULL.
<code>weights</code>	(numeric, n) a vector of weights, may be NULL.
<code>nIter, burnIn, thin</code>	(integer) the number of iterations, burn-in and thinning.
<code>saveAt</code>	(string) this may include a path and a pre-fix that will be added to the name of the files that are saved as the program runs.
<code>prior</code>	(list) containing the following elements, <ul style="list-style-type: none"> • <code>prior\$varE</code>, <code>prior\$varBR</code>, <code>prior\$varU</code>: (list) each providing degree of freedom (<code>\$df</code>) and scale (<code>\$S</code>). These are the parameters of the scaled inverse-χ^2 distributions assigned to variance components, see Eq. (2) below. In the parameterization used by <code>BLR()</code> the prior expectation of variance parameters is $S/(df - 2)$. • <code>prior\$lambda</code>: (list) providing <code>\$value</code> (initial value for λ); <code>\$type</code> ('random' or 'fixed') this argument specifies whether λ should be kept fixed at the value provided by <code>\$value</code> or updated with samples from the posterior distribution; and, either <code>\$shape</code> and <code>\$rate</code> (this when a Gamma prior is desired on λ^2) or <code>\$shape1</code>, <code>\$shape2</code> and <code>\$max</code>, in this case $p(\lambda \max, \alpha_1, \alpha_2) \propto \text{Beta}\left(\frac{\lambda}{\max} \alpha_1, \alpha_2\right)$. For detailed description of these priors see de los Campos <i>et al.</i> (2009).
<code>thin2</code>	This value controls whether the running means are saved to disk or not. If <code>thin2</code> is greater than <code>nIter</code> the running means are not saved (default, <code>thin2=1 \times 10^{10}</code>).
<code>minAbsBeta</code>	The minimum absolute value of the components of β_L to avoid numeric problems when sampling from τ^2 , default 1×10^{-9}

Details

The program runs a Gibbs sampler for the Bayesian regression model described below.

Likelihood. The equation for the data is:

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{X}_F\beta_F + \mathbf{X}_R\beta_R + \mathbf{X}_L\beta_L + \mathbf{Z}\mathbf{u} + \boldsymbol{\varepsilon} \quad (1)$$

where `y`, the response is a $n \times 1$ vector (NAs allowed); μ is an intercept; \mathbf{X}_F , \mathbf{X}_R , \mathbf{X}_L and \mathbf{Z} are incidence matrices used to accommodate different types of effects (see below), and; $\boldsymbol{\varepsilon}$ is a vector of

model residuals assumed to be distributed as $\varepsilon \sim N(\mathbf{0}, \text{Diag}(\sigma_\varepsilon^2/w_i^2))$, here σ_ε^2 is an (unknown) variance parameter and w_i are (known) weights that allow for heterogeneous-residual variances.

Any of the elements in the right-hand side of the linear predictor, except μ and ε , can be omitted; by default the program runs an intercept model.

Prior. The residual variance is assigned a scaled inverse- χ^2 prior with degree of freedom and scale parameter provided by the user, that is, $\sigma_\varepsilon^2 \sim \chi^{-2}(\sigma_\varepsilon^2|df_\varepsilon, S_\varepsilon)$. The regression coefficients $\{\mu, \beta_F, \beta_R, \beta_L, \mathbf{u}\}$ are assigned priors that yield different type of shrinkage. The intercept and the vector of regression coefficients β_F are assigned flat priors (i.e., estimates are not shrunk). The vector of regression coefficients β_R is assigned a Gaussian prior with variance common to all effects, that is, $\beta_{R,j} \stackrel{iid}{\sim} N(0, \sigma_{\beta_R}^2)$. This prior is the Bayesian counterpart of Ridge Regression. The variance parameter $\sigma_{\beta_R}^2$, is treated as unknown and it is assigned a scaled inverse- χ^2 prior, that is, $\sigma_{\beta_R}^2 \sim \chi^{-2}(\sigma_{\beta_R}^2|df_{\beta_R}, S_{\beta_R})$ with degrees of freedom df_{β_R} , and scale S_{β_R} provided by the user.

The vector of regression coefficients β_L is treated as in the Bayesian LASSO of Park and Casella (2008). Specifically,

$$p(\beta_L, \tau^2, \lambda | \sigma_\varepsilon^2) = \left\{ \prod_k N(\beta_{L,k} | 0, \sigma_\varepsilon^2 \tau_k^2) \text{Exp}(\tau_k^2 | \lambda^2) \right\} p(\lambda),$$

where, $\text{Exp}(\cdot)$ is an exponential prior and $p(\lambda)$ can either be: (a) a mass-point at some value (i.e., fixed λ); (b) $p(\lambda^2) \sim \text{Gamma}(r, \delta)$ this is the prior suggested by Park and Casella (2008); or, (c) $p(\lambda | \max, \alpha_1, \alpha_2) \propto \text{Beta}(\frac{\lambda}{\max} | \alpha_1, \alpha_2)$, see de los Campos *et al.* (2009) for details. It can be shown that the marginal prior of regression coefficients $\beta_{L,k}$, $\int N(\beta_{L,k} | 0, \sigma_\varepsilon^2 \tau_k^2) \text{Exp}(\tau_k^2 | \lambda^2) \partial \tau_k^2$, is Double-Exponential. This prior has thicker tails and higher peak of mass at zero than the Gaussian prior used for β_R , inducing a different type of shrinkage.

The vector \mathbf{u} is used to model the so called ‘infinitesimal effects’, and is assigned a prior $\mathbf{u} \sim N(\mathbf{0}, \mathbf{A}\sigma_u^2)$, where, \mathbf{A} is a positive-definite matrix (usually a relationship matrix computed from a pedigree) and σ_u^2 is an unknown variance, whose prior is $\sigma_u^2 \sim \chi^{-2}(\sigma_u^2|df_u, S_u)$.

Collecting the above mentioned assumptions, the posterior distribution of model unknowns, $\theta = \{\mu, \beta_F, \beta_R, \sigma_{\beta_R}^2, \beta_L, \tau^2, \lambda, \mathbf{u}, \sigma_u^2, \sigma_\varepsilon^2\}$, is,

$$\begin{aligned} p(\theta | \mathbf{y}) &\propto N\left(\mathbf{y} | \mathbf{1}\mu + \mathbf{X}_F\beta_F + \mathbf{X}_R\beta_R + \mathbf{X}_L\beta_L + \mathbf{Z}\mathbf{u}; \text{Diag}\left\{\frac{\sigma_\varepsilon^2}{w_i^2}\right\}\right) \\ &\times \left\{ \prod_j N(\beta_{R,j} | 0, \sigma_{\beta_R}^2) \right\} \chi^{-2}(\sigma_{\beta_R}^2 | df_{\beta_R}, S_{\beta_R}) \\ &\times \left\{ \prod_k N(\beta_{L,k} | 0, \sigma_\varepsilon^2 \tau_k^2) \text{Exp}(\tau_k^2 | \lambda^2) \right\} p(\lambda) \\ &\times N(\mathbf{u} | \mathbf{0}, \mathbf{A}\sigma_u^2) \chi^{-2}(\sigma_u^2 | df_u, S_u) \chi^{-2}(\sigma_\varepsilon^2 | df_\varepsilon, S_\varepsilon) \end{aligned} \quad (2)$$

Value

A list with posterior means, posterior standard deviations, and the parameters used to fit the model:

\$yHat	the posterior mean of $\mathbf{1}\mu + \mathbf{X}_F\beta_F + \mathbf{X}_R\beta_R + \mathbf{X}_L\beta_L + \mathbf{Z}\mathbf{u} + \varepsilon$.
\$SD.yHat	the corresponding posterior standard deviation.
\$mu	the posterior mean of the intercept.

\$varE	the posterior mean of σ_{ϵ}^2 .
\$bR	the posterior mean of β_R .
\$SD.bR	the corresponding posterior standard deviation.
\$varBr	the posterior mean of $\sigma_{\beta_R}^2$.
\$bL	the posterior mean of β_L .
\$SD.bL	the corresponding posterior standard deviation.
\$tau2	the posterior mean of τ^2 .
\$lambda	the posterior mean of λ .
\$u	the posterior mean of u .
\$SD.u	the corresponding posterior standard deviation.
\$varU	the posterior mean of σ_u^2 .
\$fit	a list with evaluations of effective number of parameters and DIC (Spiegelhalter <i>et al.</i> , 2002).
\$whichNa	a vector indicating which entries in y were missing.
\$prior	a list containing the priors used during the analysis.
\$weights	vector of weights.
\$fit	list containing the following elements, <ul style="list-style-type: none"> • \$logLikAtPostMean: log-likelihood evaluated at posterior mean. • \$postMeanLogLik: the posterior mean of the Log-Likelihood. • \$pD: estimated effective number of parameters, Spiegelhalter <i>et al.</i> (2002). • \$DIC: the deviance information criterion, Spiegelhalter <i>et al.</i> (2002).
\$nIter	the number of iterations made in the Gibbs sampler.
\$burnIn	the number of iterations used as burn-in.
\$thin	the thin used.
\$y	original data-vector.

The posterior means returned by BLR are calculated after burnIn is passed and at a thin as specified by the user.

Save. The routine will save samples of μ , variance components and λ and running means (rm*.dat). Running means are computed using the thinning specified by the user (see argument thin above); however these running means are saved at a thinning specified by argument thin2 (by default, thin2=1 \times 10¹⁰ so that running means are computed as the sampler runs but not saved to the disc).

Author(s)

Gustavo de los Campos, Paulino Perez Rodriguez,

References

- de los Campos G., H. Naya, D. Gianola, J. Crossa, A. Legarra, E. Manfredi, K. Weigel and J. Cotes. 2009. Predicting Quantitative Traits with Regression Models for Dense Molecular Markers and Pedigree. *Genetics* **182**: 375-385.
- Park T. and G. Casella. 2008. The Bayesian LASSO. *Journal of the American Statistical Association* **103**: 681-686.
- Spiegelhalter, D.J., N.G. Best, B.P. Carlin and A. van der Linde. 2002. Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* **64** (4): 583-639.

Examples

```
## Not run:
#####
##Example 1:
#####

rm(list=ls())
setwd(tempdir())
library(BGLR)
data(wheat)      #Loads the wheat dataset

y=wheat.Y[,1]
### Creates a testing set with 100 observations
whichNa<-sample(1:length(y),size=100,replace=FALSE)
yNa<-y
yNa[whichNa]<-NA

### Runs the Gibbs sampler
fm<-BLR(y=yNa,XL=wheat.X,GF=list(ID=1:nrow(wheat.A),A=wheat.A),
      prior=list(varE=list(df=3,S=0.25),
      varU=list(df=3,S=0.63),
      lambda=list(shape=0.52,rate=1e-4,
      type='random',value=30)),
      nIter=5500,burnIn=500,thin=1)

MSE.tst<-mean((fm$yHat[whichNa]-y[whichNa])^2)
MSE.tst
MSE.trn<-mean((fm$yHat[-whichNa]-y[-whichNa])^2)
MSE.trn
COR.tst<-cor(fm$yHat[whichNa],y[whichNa])
COR.tst
COR.trn<-cor(fm$yHat[-whichNa],y[-whichNa])
COR.trn

plot(fm$yHat~y,xlab="Phenotype",
      ylab="Pred. Gen. Value" ,cex=.8)
points(x=y[whichNa],y=fm$yHat[whichNa],col=2,cex=.8,pch=19)

x11()
plot(scan('varE.dat'),type="o",
```

```

        ylab=expression(paste(sigma[epsilon]^2))

#####
#Example 2: Ten fold, Cross validation, environment 1,
#####

rm(list=ls())
setwd(tempdir())
library(BGLR)
data(wheat)      #Loads the wheat dataset
nIter<-1500      #For real data sets more samples are needed
burnIn<-500
thin<-10
folds<-10
y<-wheat.Y[,1]
A<-wheat.A

priorBL<-list(
  varE=list(df=3,S=2.5),
  varU=list(df=3,S=0.63),
  lambda = list(shape=0.52,rate=1e-5,value=20,type='random')
)

set.seed(123) #Set seed for the random number generator
sets<-rep(1:10,60)[-1]
sets<-sets[order(runif(nrow(A)))]
COR.CV<-rep(NA,times=(folds+1))
names(COR.CV)<-c(paste('fold=',1:folds,sep=''),'Pooled')
w<-rep(1/nrow(A),folds) ## weights for pooled correlations and MSE
yHatCV<-numeric()

for(fold in 1:folds)
{
  yNa<-y
  whichNa<-which(sets==fold)
  yNa[whichNa]<-NA
  prefix<-paste('PM_BL','_fold_',fold,'_',sep='')
  fm<-BLR(y=yNa,XL=wheat.X,GF=list(ID=(1:nrow(wheat.A)),A=wheat.A),prior=priorBL,
    nIter=nIter,burnIn=burnIn,thin=thin)
  yHatCV[whichNa]<-fm$yHat[fm$whichNa]
  w[fold]<-w[fold]*length(fm$whichNa)
  COR.CV[fold]<-cor(fm$yHat[fm$whichNa],y[whichNa])
}

COR.CV[11]<-mean(COR.CV[1:10])
COR.CV

#####

## End(Not run)

```

getVariances	<i>getVariances</i>
--------------	---------------------

Description

Computes the sample-variance (var()) for sets of markers as well as the total variance.

Usage

```
getVariances(X, B, sets, verbose=TRUE)
```

Arguments

X	(numeric, $n \times p$) incidence matrix for β .
B	(numeric), object returned by the function readBinMat().
sets	(numeric).
verbose	(logical), if TRUE it shows progress information in the console.

Value

A matrix with variances for markers as well as the total.

Author(s)

Gustavo de los Campos.

Examples

```
## Not run:
#Demos

library(BGLR)
data(wheat)
y=wheat.Y[,1] ; X=scale(wheat.X)
dir.create('test_saveEffects')
setwd('test_saveEffects')
fm=BGLR(y=y,ETA=list(list(X=X,model='BayesB',saveEffects=TRUE)),nIter=12000,thin=2,burnIn=2000)
B=readBinMat('ETA_1_b.bin')
plot(B[,1],type='o',col=4)
VAR=getVariances(B=B,X=X,sets=sample(1:20,size=1279,replace=T))
head(VAR)
plot(VAR[,"total"])

## End(Not run)
```

mice

mice dataset

Description

The mice data comes from an experiment carried out to detect and locate QTLs for complex traits in a mice population (Valdar et al. 2006a; 2006b). This data has already been analyzed for comparing genome-assisted genetic evaluation methods (Legarra et al. 2008). The data file consists of 1814 individuals, each genotyped for 10,346 polymorphic markers. The trait here here is body mass index (BMI), and additional information about body weight, season, month and day.

Usage

```
data(mice)
```

Format

Matrix mice.A contains the pedigree. The matrix mice.X contains the marks information and mice.pheno contains phenotypical information.

References

- Legarra A., Robert-Granie, E. Manfredi, and J. M. Elsen, 2008 Performance of genomic selection in mice. *Genetics* 180:611-618.
- Valdar, W., L. C. Solberg, D. Gauguier, S. Burnett, P. Klenerman et al., 2006a Genome-wide genetic association of complex traits in heterogeneous stock mice. *Nat. Genet.* 38:879-887.
- Valdar, W., L. C. Solberg, D. Gauguier, W. O. Cookson, J. N. P. Rawlis et al., 2006b Genetic and environmental effects on complex traits in mice. *Genetics*, 174:959-984.

mice.A

Pedigree info for the mice dataset

Description

Is a numerator relationship matrix (1814 x 1814) computed from a pedigree that traced back many generations.

References

- de los Campos G., H. Naya, D. Gianola, J. Crossa, A. Legarra, E. Manfredi, K. Weigel and J. Cotes. 2009. Predicting Quantitative Traits with Regression Models for Dense Molecular Markers and Pedigree. *Genetics* **182**: 375-385.

mice.pheno	<i>Phenotypical data for the mice dataset</i>
------------	---

Description

A data frame with pheotypical information related to diabetes. The data frame has several columns: SUBJECT.NAME, PROJECT.NAME, PHENOTYPE.NAME, Obesity.BMI, Obesity.BodyLength, Date.Month, Date.Year, Date.Season,cDate.StudyStartSeconds, Date.Hour, Date.StudyDay, GENDER, EndNormalBW, CoatColour, CageDensity, Litter, cage.

The phenotypes are described in <http://gscan.well.ox.ac.uk>.

mice.X	<i>Molecular markers</i>
--------	--------------------------

Description

Is a matrix (1814 x 10346) with SNP markers.

plot.BGLR	<i>Plots for BGLR Analysis</i>
-----------	--------------------------------

Description

Plots observed vs predicted values for objects of class BGLR.

Usage

```
## S3 method for class 'BGLR'
plot(x, ...)
```

Arguments

x	An object of class BGLR.
...	Further arguments passed to or from other methods.

Author(s)

Gustavo de los Campos, Paulino Perez Rodriguez,

See Also

BGLR.

Examples

```
## Not run:  
  
setwd(tempdir())  
library(BGLR)  
data(wheat)  
out=BLR(y=wheat.Y[,1],XL=wheat.X)  
plot(out)  
  
## End(Not run)
```

predict.BGLR

Model Predictions

Description

extracts predictions from the results of BGLR function.

Usage

```
## S3 method for class 'BGLR'  
predict(object, newdata, ...)
```

Arguments

object	An object of class BGLR.
newdata	Currently not supported, for new data you should assing missing value indicator (NAs) to the corresponding entries in the response vector (y).
...	Further arguments passed to or from other methods.

Author(s)

Gustavo de los Campos, Paulino Perez Rodriguez,

See Also

BGLR.

Examples

```
## Not run:

setwd(tempdir())
library(BGLR)
data(wheat)
out=BLR(y=wheat.Y[,1],XL=wheat.X)
predict(out)

## End(Not run)
```

readBinMat

readBinMat

Description

Function to read effects saved by BGLR when `ETA[[j]]$saveEffects=TRUE`.

Usage

```
readBinMat(filename, byrow=TRUE)
```

Arguments

`filename` (string), the name of the file to be read.
`byrow` (logical), if TRUE the matrix is created by filling its corresponding elements by rows.

Value

A matrix with samples of regression coefficients.

Author(s)

Gustavo de los Campos.

Examples

```
## Not run:
#Demos

library(BGLR)
data(wheat)
```

```

y=wheat.Y[,1] ; X=scale(wheat.X)
dir.create('test_saveEffects')
setwd('test_saveEffects')
fm=BGLR(y=y,ETA=list(list(X=X,model='BayesB',saveEffects=TRUE)),nIter=12000,thin=2,burnIn=2000)
B=readBinMat('ETA_1_b.bin')

## End(Not run)

```

read_bed

read_bed

Description

This function reads genotype information stored in binary PED (BED) files used in plink. These files save space and time. The pedigree/phenotype information is stored in a separate file (*.fam) and the map information is stored in an extended MAP file (*.bim) that contains information about the allele names, which would otherwise be lost in the BED file. More details <http://zzz.bwh.harvard.edu/plink/binary.shtml>.

Usage

```
read_bed.bed_file,bim_file,fam_file,na.strings,verbose)
```

Arguments

bed_file	binary file with genotype information.
bim_file	text file with pedigree/phenotype information.
fam_file	text file with extended map information.
na.strings	missing value indicators, default=c("0","-9").
verbose	logical, if true print hex dump of bed file.

Value

The routine will return a vector of dimension $n \times p$ (n =number of individuals, p =number of snps), with the snps(individuals) stacked, depending whether the BED file is in SNP-major or individual-major mode.

The vector contains integer codes:

Integer code	Genotype
0	00 Homozygote "1"/"1"
1	01 Heterozygote
2	10 Missing genotype
3	11 Homozygote "2"/"2"

Author(s)

Gustavo de los Campos, Paulino Perez Rodriguez,

Examples

```
## Not run:

library(BGLR)
demo(read_ped)

## End(Not run)
```

read_ped	<i>read_ped</i>
----------	-----------------

Description

This function reads genotype information stored in PED format used in plink.

Usage

```
read_ped(ped_file)
```

Arguments

ped_file ASCII file with genotype information.

Details

The PED file is a white-space (space or tab) delimited file: the first six columns are mandatory:

Family ID Individual ID Paternal ID Maternal ID Sex (1=male; 2=female; other=unknown) Phenotype

The IDs are alphanumeric: the combination of family and individual ID should uniquely identify a person. A PED file must have 1 and only 1 phenotype in the sixth column. The phenotype can be either a quantitative trait or an affection status column.

Value

The routine will return a vector of dimension $n \times p$ (n =number of individuals, p =number of snps), with the snps stacked.

The vector contains integer codes:

Integer code	Genotype
0	00 Homozygote "1"/"1"

1	01 Heterozygote
2	10 Missing genotype
3	11 Homozygote "2"/"2"

Author(s)

Gustavo de los Campos, Paulino Perez Rodriguez,

Examples

```
## Not run:  
  
library(BGLR)  
demo(read_ped)  
  
## End(Not run)
```

residuals.BGLR	<i>Extracts models residuals</i>
----------------	----------------------------------

Description

extracts model residuals from objects returned by BGLR function.

Usage

```
## S3 method for class 'BGLR'  
residuals(object, ...)
```

Arguments

object	An object of class BGLR.
...	Further arguments passed to or from other methods.

Author(s)

Gustavo de los Campos, Paulino Perez Rodriguez,

See Also

BGLR.

Examples

```
## Not run:

setwd(tempdir())
library(BGLR)
data(wheat)
out=BLR(y=wheat.Y[,1],XL=wheat.X)
residuals(out)

## End(Not run)
```

summary.BGLR

summary for BGLR fitted models

Description

Gives a summary for a fitted model using BGLR function.

Usage

```
## S3 method for class 'BGLR'
summary(object, ...)
```

Arguments

object An object of class BGLR.
... Further arguments passed to or from other methods.

Author(s)

Gustavo de los Campos, Paulino Perez Rodriguez,

See Also

BGLR.

Examples

```
## Not run:

setwd(tempdir())
library(BGLR)
data(wheat)
out=BLR(y=wheat.Y[,1],XL=wheat.X)
summary(out)
```

```
## End(Not run)
```

wheat

wheat dataset

Description

Information from a collection of 599 historical CIMMYT wheat lines. The wheat data set is from CIMMYT's Global Wheat Program. Historically, this program has conducted numerous international trials across a wide variety of wheat-producing environments. The environments represented in these trials were grouped into four basic target sets of environments comprising four main agro-climatic regions previously defined and widely used by CIMMYT's Global Wheat Breeding Program. The phenotypic trait considered here was the average grain yield (GY) of the 599 wheat lines evaluated in each of these four mega-environments.

A pedigree tracing back many generations was available, and the Browse application of the International Crop Information System (ICIS), as described in <http://repository.cimmyt.org/xmlui/bitstream/handle/10883/3488/72673.pdf> (McLaren *et al.* 2000, 2005) was used for deriving the relationship matrix A among the 599 lines; it accounts for selection and inbreeding.

Wheat lines were recently genotyped using 1447 Diversity Array Technology (DArT) generated by Triticaret Pty. Ltd. (Canberra, Australia; <https://www.diversityarrays.com>). The DArT markers may take on two values, denoted by their presence or absence. Markers with a minor allele frequency lower than 0.05 were removed, and missing genotypes were imputed with samples from the marginal distribution of marker genotypes, that is, $x_{ij} = \text{Bernoulli}(\hat{p}_j)$, where \hat{p}_j is the estimated allele frequency computed from the non-missing genotypes. The number of DArT MMs after edition was 1279.

Usage

```
data(wheat)
```

Format

Matrix Y contains the average grain yield, column 1: Grain yield for environment 1 and so on. The matrix A contains additive relationship computed from the pedigree and matrix X contains the markers information.

Source

International Maize and Wheat Improvement Center (CIMMYT), Mexico.

References

McLaren, C. G., L. Ramos, C. Lopez, and W. Eusebio. 2000. "Applications of the genealogy management system." In *International Crop Information System. Technical Development Manual, version VI*, edited by McLaren, C. G., J.W. White and P.N. Fox. pp. 5.8-5.13. CIMMYT, Mexico: CIMMYT and IRRI.

McLaren, C. G., R. Bruskiewich, A.M. Portugal, and A.B. Cosico. 2005. The International Rice Information System. A platform for meta-analysis of rice crop data. *Plant Physiology* **139**: 637-642.

wheat.A

Pedigree info for the wheat dataset

Description

Is a numerator relationship matrix (599 x 599) computed from a pedigree that traced back many generations. This relationship matrix was derived using the Browse application of the International Crop Information System (ICIS), as described in <http://repository.cimmyt.org/xmlui/bitstream/handle/10883/3488/72673.pdf> (McLaren *et al.* 2000, 2005).

Source

International Maize and Wheat Improvement Center (CIMMYT), Mexico.

References

McLaren, C. G., L. Ramos, C. Lopez, and W. Eusebio. 2000. "Applications of the genealogy management system." In *International Crop Information System. Technical Development Manual, version VI*, edited by McLaren, C. G., J.W. White and P.N. Fox. pp. 5.8-5.13. CIMMYT, Mexico: CIMMYT and IRRI.

McLaren, C. G., R. Bruskiewich, A.M. Portugal, and A.B. Cosico. 2005. The International Rice Information System. A platform for meta-analysis of rice crop data. *Plant Physiology* **139**: 637-642.

wheat.sets

Sets for cross validation (CV)

Description

Is a vector (599 x 1) that assigns observations to 10 disjoint sets; the assignment was generated at random. This is used later to conduct a 10-fold CV.

Source

International Maize and Wheat Improvement Center (CIMMYT), Mexico.

wheat.X

Molecular markers

Description

Is a matrix (599 x 1279) with DArT genotypes; data are from pure lines and genotypes were coded as 0/1 denoting the absence/presence of the DArT. Markers with a minor allele frequency lower than 0.05 were removed, and missing genotypes were imputed with samples from the marginal distribution of marker genotypes, that is, $x_{ij} = \text{Bernoulli}(\hat{p}_j)$, where \hat{p}_j is the estimated allele frequency computed from the non-missing genotypes. The number of DArT MMs after edition was 1279.

Source

International Maize and Wheat Improvement Center (CIMMYT), Mexico.

wheat.Y

Grain yield

Description

A matrix (599 x 4) containing the 2-yr average grain yield of each of these lines in each of the four environments (phenotypes were standardized to a unit variance within each environment).

Source

International Maize and Wheat Improvement Center (CIMMYT), Mexico.

write_bed

write_bed

Description

This function writes genotype information into a binary PED (BED) file used in plink. For more details about this format see <http://zzz.bwh.harvard.edu/plink/binary.shtml>.

Usage

```
write_bed(x,n,p,bed_file)
```

Arguments

n	integer, number of individuals.
p	integer, number of SNPs.
x	integer vector that contains the genotypic information coded as 0,1,2 and 3 (see details below). The information must be in snp major order. The vector should be of dimension n*p with the snps stacked.
bed_file	output binary file with genotype information.

Details

The vector contains integer codes:

Integer code	Genotype
0	00 Homozygote "1"/"1"
1	01 Heterozygote
2	10 Missing genotype
3	11 Homozygote "2"/"2"

Author(s)

Gustavo de los Campos, Paulino Perez Rodriguez,

Examples

```
## Not run:  
  
library(BGLR)  
demo(write_bed)  
  
## End(Not run)
```

Index

*Topic **datasets**

mice, [12](#)
mice.A, [12](#)
mice.pheno, [13](#)
mice.X, [13](#)
wheat, [20](#)
wheat.A, [21](#)
wheat.sets, [21](#)
wheat.X, [22](#)
wheat.Y, [22](#)

*Topic **models**

BGLR, [2](#)
BLR, [5](#)
getVariances, [11](#)
readBinMat, [15](#)

*Topic **plot**

plot.BGLR, [13](#)

*Topic **residuals**

residuals.BGLR, [18](#)

*Topic **summary**

summary.BGLR, [19](#)

BGLR, [2](#)

BLR, [5](#)

getVariances, [11](#)

mice, [12](#)

mice.A, [12](#)

mice.pheno, [13](#)

mice.X, [13](#)

plot.BGLR, [13](#)

predict.BGLR, [14](#)

read_bed, [16](#)

read_ped, [17](#)

readBinMat, [15](#)

residuals.BGLR, [18](#)

summary.BGLR, [19](#)

wheat, [20](#)

wheat.A, [21](#)

wheat.sets, [21](#)

wheat.X, [22](#)

wheat.Y, [22](#)

write_bed, [22](#)