Package 'BGLR'

October 7, 2013

Version 1.0

Date 2012-09-12

Title Bayesian Generalized Linear Regression

Author Gustavo de los Campos, Paulino Perez Rodriguez,

Maintainer Paulino Perez Rodriguez <perpdgo@colpos.mx>

Depends R (>= 2.12.2)

Description Bayesian Generalized Linear Regression

LazyLoad true

License GPL-3

R topics documented:

BGLR 2
BLR 5
mice 10
mice.A 10
mice.pheno 11
mice.X 11
plot.BGLR 11
predict.BGLR 12
read_bed 13
read_ped 14
wheat 15
wheat.A 16
wheat.sets 16
wheat.X 17
wheat.Y 17
write_bed 17

Index **19**

BGLR *Bayesian Generalized Linear Regression*

Description

The BGLR ('Bayesian Generalized Linear Regression') function fits various types of parametric and semi-parametric Bayesian regressions to continuos (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, minAbsBeta = 1e-09, weights = NULL,
     verbose = TRUE, rmExistingFiles = TRUE)
```

Arguments

y	(numeric, n) the data-vector (NAs allowed).
response_type	(string) admits values "gausian" 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 ot the outcomes is determined based on the alphanumeric order (0<1<2..<a<b..). For ordinal traits the probit link is used.
a,b	(numeric, n) only requiered for censored Gaussian 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, the following 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: 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 sclaed-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. X~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 detail about the models in BGLR see the vignettes in the package or http://genomics.cimmyt.org/BGLR.pdf.
weights	(numeric, n) 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 square of the 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 detauls see the vignettes in the package or http://genomics.cimmyt.org/BGLR.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, internaly, hyper-paramters 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.pdf.

`verbose`
: (logical) if TRUE the iteration history is printed, default TRUE.

`minAbsBeta`
: (numeric) The minimum absolute value of the components of $\boldsymbol{\beta}_L$ to avoid numeric problems when sampling from $\boldsymbol{\tau}^2$, default 1×10^{-9}.

`rmExistingFiles`
: (logical) if TRUE removes existing output files from previous runs, default TRUE.

Details

BGLR implements a Gibbs sampler for a Bayesian regresion 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:

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

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 an upper vound 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) the threshold is set equal to zero, for more than two classess thresholds are estimated from the data. Further details about this approach can be found in Albert and Chib (1993).

Value

A list with estunated posterior means, estiamted posterior standard deviations, and the parameters used to fit the model. See the vignettes in the package (or http://genomics.cimmyt.org/BGLR.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

y	(numeric, n) the data-vector (NAs allowed).
XF	(numeric, $n \times pF$) incidence matrix for $\boldsymbol{\beta}_F$, may be NULL.
XR	(numeric, $n \times pR$) incidence matrix for $\boldsymbol{\beta}_R$, may be NULL.
XL	(numeric, $n \times pL$) incidence matrix for $\boldsymbol{\beta}_L$, may be NULL.
GF	(list) providing an \$ID (integer, n) linking observations to groups (e.g., lines or sires) and a (co)variance structure (\$A, numeric, $pU \times pU$) between effects of the grouping factor (e.g., line or sire effects). Note: ID must be an integer taking values from 1 to pU; ID[i]=q indicates that the ith observation in $\boldsymbol{y}$ belongs to cluster q whose (co)variance function is in the qth row (column) of $\boldsymbol{A}$. GF may be NULL.
weights	(numeric, n) a vector of weights, may be NULL.
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.
prior	(list) containing the following elements, • prior\$varE, prior\$varBR, prior\$varU: (list) each providing degree of freedom (\$df) and scale (\$S). These are the parameters of the scaled inverse-χ^2 distributions assigned to variance components, see Eq. (2) below. In the parameterization used by BLR() the prior expectation of variance parameters is $S/(df-2)$.

- prior$lambda: (list) providing $value (initial value for λ); $type ('random' or 'fixed') this argument specifies whether λ should be kept fixed at the value provided by $value or updated with samples from the posterior distribution; and, either $shape and $rate (this when a Gamma prior is desired on λ^2) or $shape1, $shape2 and $max, in this case $p(\lambda|\max,\alpha_1,\alpha_2) \propto Beta\left(\frac{\lambda}{\max}|\alpha_1,\alpha_2\right)$. For detailed description of these priors see de los Campos *et al.* (2009).

`thin2` This value controls wether the running means are saved to disk or not. If thin2 is greater than nIter the running means are not saved (default, thin2=1×10^{10}).

`minAbsBeta` The minimum absolute value of the components of $\boldsymbol{\beta}_L$ to avoid numeric problems when sampling from $\boldsymbol{\tau}^2$, 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:

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

where $\boldsymbol{y}$, the response is a $n \times 1$ vector (NAs allowed); μ is an intercept; $\boldsymbol{X}_F, \boldsymbol{X}_R, \boldsymbol{X}_L$ and $\boldsymbol{Z}$ 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 $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, Diag(\sigma^2_{\boldsymbol{\varepsilon}}/w_i^2))$, here $\sigma^2_{\boldsymbol{\varepsilon}}$ 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 $\boldsymbol{\varepsilon}$, 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, $\sigma^2_{\boldsymbol{\varepsilon}} \sim \chi^{-2}(\sigma^2_{\boldsymbol{\varepsilon}}|df_{\boldsymbol{\varepsilon}}, S_{\boldsymbol{\varepsilon}})$. The regression coefficients $\{\mu, \boldsymbol{\beta}_F, \boldsymbol{\beta}_R, \boldsymbol{\beta}_L, \boldsymbol{u}\}$ are assigned priors that yield different type of shrinkage. The intercept and the vector of regression coefficients $\boldsymbol{\beta}_F$ are assigned flat priors (i.e., estimates are not shrunk). The vector of regression coefficients $\boldsymbol{\beta}_R$ is assigned a Gaussian prior with variance common to all effects, that is, $\beta_{R,j} \overset{iid}{\sim} N(0, \sigma^2_{\boldsymbol{\beta}_R})$. This prior is the Bayesian counterpart of Ridge Regression. The variance parameter $\sigma^2_{\boldsymbol{\beta}_R}$, is treated as unknown and it is assigned a scaled inverse-χ^2 prior, that is, $\sigma^2_{\boldsymbol{\beta}_R} \sim \chi^{-2}(\sigma^2_{\boldsymbol{\beta}_R}|df_{\boldsymbol{\beta}_R}, S_{\boldsymbol{\beta}_R})$ with degrees of freedom $df_{\boldsymbol{\beta}_R}$, and scale $S_{\boldsymbol{\beta}_R}$ provided by the user.

The vector of regression coefficients $\boldsymbol{\beta}_L$ is treated as in the Bayesian LASSO of Park and Casella (2008). Specifically,

$$p(\boldsymbol{\beta}_L, \boldsymbol{\tau}^2, \lambda|\sigma^2_{\boldsymbol{\varepsilon}}) = \left\{\prod_k N(\beta_{L,k}|0, \sigma^2_{\boldsymbol{\varepsilon}}\tau^2_k) Exp\left(\tau^2_k|\lambda^2\right)\right\} p(\lambda),$$

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

The vector $\boldsymbol{u}$ is used to model the so called 'infinitesimal effects', and is assigned a prior $\boldsymbol{u} \sim N(\mathbf{0}, \boldsymbol{A}\sigma^2_{\boldsymbol{u}})$, where, $\boldsymbol{A}$ is a positive-definite matrix (usually a relationship matrix computed from a pedigree) and $\sigma^2_{\boldsymbol{u}}$ is an unknow variance, whose prior is $\sigma^2_{\boldsymbol{u}} \sim \chi^{-2}(\sigma^2_{\boldsymbol{u}}|df_{\boldsymbol{u}}, S_{\boldsymbol{u}})$.

Collecting the above mentioned assumptions, the posterior distribution of model unknowns, $\boldsymbol{\theta} = \left\{\mu, \boldsymbol{\beta}_F, \boldsymbol{\beta}_R, \sigma^2_{\boldsymbol{\beta}_R}, \boldsymbol{\beta}_L, \boldsymbol{\tau}^2, \lambda, \boldsymbol{u}, \sigma^2_{\boldsymbol{u}}, \sigma^2_{\boldsymbol{\varepsilon}}, \right\}$, is,

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{y}) \quad \propto \quad & N\left(\boldsymbol{y}|\mathbf{1}\mu + \boldsymbol{X}_F\boldsymbol{\beta}_F + \boldsymbol{X}_R\boldsymbol{\beta}_R + \boldsymbol{X}_L\boldsymbol{\beta}_L + \boldsymbol{Z}\boldsymbol{u}; Diag\left\{\frac{\sigma^2_{\boldsymbol{\varepsilon}}}{w_i^2}\right\}\right) \\ & \times \left\{\prod_j N\left(\beta_{R,j}|0, \sigma^2_{\boldsymbol{\beta}_R}\right)\right\} \chi^{-2}\left(\sigma^2_{\boldsymbol{\beta}_R}|df_{\boldsymbol{\beta}_R}, S_{\boldsymbol{\beta}_R}\right) \\ & \times \left\{\prod_k N\left(\beta_{L,k}|0, \sigma^2_{\boldsymbol{\varepsilon}}\tau^2_k\right) Exp\left(\tau^2_k|\lambda^2\right)\right\} p(\lambda) \\ & \times N(\boldsymbol{u}|\mathbf{0}, \boldsymbol{A}\sigma^2_{\boldsymbol{u}})\chi^{-2}(\sigma^2_{\boldsymbol{u}}|df_{\boldsymbol{u}}, S_{\boldsymbol{u}})\chi^{-2}(\sigma^2_{\boldsymbol{\varepsilon}}|df_{\boldsymbol{\varepsilon}}, S_{\boldsymbol{\varepsilon}}) \end{aligned} \tag{2}$$

Value

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

`$yHat`	the posterior mean of $\mathbf{1}\mu + \boldsymbol{X}_F\boldsymbol{\beta}_F + \boldsymbol{X}_R\boldsymbol{\beta}_R + \boldsymbol{X}_L\boldsymbol{\beta}_L + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\varepsilon}$.
`$SD.yHat`	the corresponding posterior standard deviation.
`$mu`	the posterior mean of the intercept.
`$varE`	the posterior mean of $\sigma^2_{\boldsymbol{\varepsilon}}$.
`$bR`	the posterior mean of $\boldsymbol{\beta}_R$.
`$SD.bR`	the corresponding posterior standard deviation.
`$varBr`	the posterior mean of $\sigma^2_{\boldsymbol{\beta}_R}$.
`$bL`	the posterior mean of $\boldsymbol{\beta}_L$.
`$SD.bL`	the corresponding posterior standard deviation.
`$tau2`	the posterior mean of $\boldsymbol{\tau}^2$.
`$lambda`	the posterior mean of λ.
`$u`	the posterior mean of $\boldsymbol{u}$.
`$SD.u`	the corresponding posterior standard deviation.
`$varU`	the posterior mean of $\sigma^2_{\boldsymbol{u}}$.
`$fit`	a list with evaluations of effective number of parameters and DIC (Spiegelhalter *et al.*, 2002).
`$whichNa`	a vector indicating which entries in $\boldsymbol{y}$ were missing.
`$prior`	a list containig the priors used during the analysis.
`$weights`	vector of weights.
`$fit`	list containing the following elements, • $logLikAtPostMean: log-likelihood evaluated at posterior mean. • $postMeanLogLik: the posterior mean of the Log-Likelihood. • $pD: estimated effective number of parameters, Spiegelhalter *et al.* (2002). • $DIC: the deviance information criterion, Spiegelhalter *et al.* (2002).
`$nIter`	the number of iterations made in the Gibbs sampler.
`$burnIn`	the nuber of iteratios used as burn-in.
`$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=1×10^{10} 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)
```

`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 markes information and mice.pheno contains phenotypical information.

Source

http://gscan.well.ox.ac.uk

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.

Source

http://gscan.well.ox.ac.uk

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 *Phenotypical data for the mice dataset*

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.

Source

http://gscan.well.ox.ac.uk

mice.X *Molecular markers*

Description

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

Source

http://gscan.well.ox.ac.uk

plot.BGLR *Plots for BGLR Analysis*

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 *Predictions from BGLR Analysis*

Description

Predicting values using results from BGLR function.

Usage

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

Arguments

object	An object of class BGLR.
newdata	new data, see BGLR function for more details.
...	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)


## 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 extededed MAP file (*.bim) that contains information about the allele names, which would otherwise be lost in the BED file. More details http://pngu.mgh.harvard.edu/~purcell/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*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_bed)


## End(Not run)
```

`read_ped` *read_ped*

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*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)
```

---

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 agroclimatic 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://cropwiki.irri.org/icis/index.php/TDM_GMS_Browse (McLaren *et al.* 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 Triticarte Pty. Ltd. (Canberra, Australia; http://www.triticarte.com.au). 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, $x_{ij} = Bernoulli(\hat{p}_j)$, 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., 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://cropwiki.irri.org/icis/index.php/TDM_GMS_Browse (McLaren *et al.* 2005).

Source

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

References

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, $x_{ij} = Bernoulli(\hat{p}_j)$, 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) filed used in plink. For more details about this format see http://pngu.mgh.harvard.edu/~purcell/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, 10
mice.A, 10
mice.pheno, 11
mice.X, 11
wheat, 15
wheat.A, 16
wheat.sets, 16
wheat.X, 17
wheat.Y, 17
∗Topic **models**
BGLR, 2
BLR, 5
∗Topic **plot**
plot.BGLR, 11
∗Topic **regression**
predict.BGLR, 12

BGLR, 2
BLR, 5

mice, 10
mice.A, 10
mice.pheno, 11
mice.X, 11

plot.BGLR, 11
predict.BGLR, 12

read_bed, 13
read_ped, 14

wheat, 15
wheat.A, 16
wheat.sets, 16
wheat.X, 17
wheat.Y, 17
write_bed, 17