Package 'BGLR'

September 12, 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 *Bayesian Generalized Linear Regression*

Description

The BGLR ('Bayesian Generalized Linear Regression') function was designed to fit parametric regression models using different types of shrinkage methods. Several of the models implemented in this function were presented in de los Campos *et al.* (2009, 2010).

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, specify the distribution of the response variable, right now only gaussian, Bernoulli and ordinal responses are allowed.
a	vector for specifying lower bound for censored observations, default value NULL. See details.
b	vector for specifying upper bound for censored observations, default value NULL. See details.
ETA	A list of predictors and prior specifications for the regression coefficients. For example the prior for the regression coefficients can be that used in Bayesian LASSO, Bayesian ridge regression, BayesA, BayesB, BayesC-pi, Elastic Net LASSO, etc. See details below.
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.
S0	The scale parameter for the scaled inverse-chi squared distribution for σ_e^2.
df0	The degrees of freedom for the scaled inverse-chi squared distribution for σ_e^2.
R2	...
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}.
verbose	logical, if TRUE prints iteration history, defalt TRUE.
rmExistingFiles	logical, if TRUE removes existing output files from previous runs, default value is TRUE.

Details

The program run a Gibbs sampler for the regression model given below.

Likelihood. The equation for the data is:

$$\boldsymbol{y} = \mathbf{1}\mu + \boldsymbol{X}_F\boldsymbol{\beta}_F + \sum_{h=1}^{H\beta}\boldsymbol{X}_{Rh}\boldsymbol{\beta}_{Rh} + \sum_{h=1}^{Hu}\boldsymbol{u}_h + \boldsymbol{\varepsilon} \quad (1)$$

where μ is an effect common to all individuals, $\boldsymbol{X}_F = \{x_{Fij}\}$ represent covariates whose effects $\boldsymbol{\beta}_F = \{\beta_{Fj}\}$ will be estimated shrinkage (the so-called 'fixed effecs', e.g., age, sex), $\boldsymbol{X}_{Rh} = \{x_{Rhij}\}$ represent covariates whose effects $\boldsymbol{\beta}_{Rh} = \{\boldsymbol{\beta}_{Rhj}\}$ will be treated as 'random effects' and will be estimated using shrinkage estimation methods (non-flat priors in a Bayesian context) and $\boldsymbol{u}_h = \{u_{hi}\}$ are random effects used to describe, for example, a regression on a pedigree or a RKHS regression on markers.

Prior

The model specification is complete once we assign a prior distribution to the model unknowns. The intercept μ and $\boldsymbol{\beta}_F$ are assigned flat priors, while $\boldsymbol{\beta}_{Rh}$, $\boldsymbol{u}_h$ and σ_e^2 are assigned non flat priors, denoted as $p(\boldsymbol{\beta}_R)$, $p(\boldsymbol{u})$ and $p(\sigma_e^2)$, respectively. The structure of the priors is as follows:

$$p(\mu, \boldsymbol{\beta}_F, \boldsymbol{\beta}_{R1}, ..., \boldsymbol{\beta}_{RH\beta}, \boldsymbol{u}_1, ..., \boldsymbol{u}_{Hu}, \sigma_e^2) \propto \left\{\prod_{h=1}^{H\beta} p(\boldsymbol{\beta}_{Rh})\right\}\left\{\prod_{h=1}^{Hu} p(\boldsymbol{u}_h)\right\}\chi^{-2}(\sigma_e^2|df, S), \quad (2)$$

where $\chi^{-2}(\sigma^2|df, S)$ is a scaled-inverse Chi-square density assigned to σ^2 with degree of freedom and scale parameter df and S respectively.

The prior distribution assigned to $p(\boldsymbol{u}_h|\boldsymbol{\theta}_{uh})$ is multivariate normal centered at zero and with covariance $\sigma_{uh}^2\boldsymbol{K}_{uh}$ where $\boldsymbol{K}_{uh}$ is a positive definite-matrix and σ_{uh}^2 is an unknow variance paramter. The prior assigned to this parameter is a scaled inverse chi-squared so that

$$p(\boldsymbol{u}_h, \sigma_{uh}^2) = N(\boldsymbol{u}_h|\mathbf{0}, \sigma_{uh}^2\boldsymbol{K}_{uh})\chi^{-2}(S_{uh}, df_{uh}) \quad (3)$$

Following standard assumptions of Bayesian regression models, regression coefficients are assigned IID priors; therefore: $p(\boldsymbol{\beta}_{Rh}|\boldsymbol{\theta}_{Rh}) = \left\{\prod_{j=1}^{p_{Rh}} p(\beta_{Rhj}|\boldsymbol{\theta}_{Rh})\right\} p(\boldsymbol{\theta}_{Rh})$, where $p(\beta_{Rhj}|\boldsymbol{\theta}_{Rh})$ can be a double exponential distribution, a normal distribution, etc., $\boldsymbol{\theta}_{Rh}$ is a vector of unknown indexing the prior density assigned to marker effects and $p(\boldsymbol{\theta}_{Rh})$ is the prior assigned to these unknowns.

Collecting assumptions we have:

$$\begin{aligned} p(\mu, \boldsymbol{\beta}_F, \boldsymbol{\beta}_{R1}, ..., \boldsymbol{\beta}_{RH\beta}, \boldsymbol{u}_1, ..., \boldsymbol{u}_{Hu}, \sigma^2) \quad & \propto \prod_{h=1}^{H\beta}\left\{\prod_{j=1}^{p_{Rh}} p(\beta_{Rhj}|\boldsymbol{\theta}_{Rh})\right\} p(\boldsymbol{\theta}_{Rh}) \\ & \times \left\{\prod_{h=1}^{Hu} N(\boldsymbol{u}_h|\mathbf{0}, \sigma_{uh}^2\boldsymbol{K}_{uh})\chi^{-2}(\sigma_{uh}^2|S_{uh}, df_{uh})\right\} \\ & \times \chi^{-2}(\sigma^2|df, S) \end{aligned}$$

Special cases

Bayesian Gaussian Regression (BGR)

A common approach in Bayesian shrinkage estimation is to assign independent and identically distributed (IDD) conditional Gaussian priors with unknown variance. This can be impemented by setting

$$p(\boldsymbol{\beta}_{Rh}|\boldsymbol{\theta}_{Rh}) = \left\{\prod_{j=1}^{p_{Rh}} N(\beta_{Rhj}|0, \sigma_{\beta h}^2)\right\}\chi^{-2}(\sigma_{\beta h}^2|df_{\beta h}, S_{\beta h}).$$

When $\sigma^2_{\beta h}$ is known, using this prior yield estimates which are equivalent to those of a RR. In a BGR the excent of shrinkage is controlled by the variance (or noise-to-signal) ratio $\lambda_h = \sigma^2_e/\sigma^2_{\beta h}$. This quantity is the same for all regression coefficients included in $\boldsymbol{\beta}_{Rh}$; this may not be appropiate if some markers are located in regions harboring QTL while others are located in regions which are not associated to genetic variance. To overcome this problem, alternative shrinkage procedures such as those described below can be used.

Mixtures of scaled-normal densities

This class of mixtures can be used as prior of marker effects to obtain a type of shrinkage different than that of a BGR. Examples of this are the double-exponential (DE) and scaled-t densities, which are commonly used as prior of maker effects in Whole Genomic Prediction (WPG). The results models are known as the Bayesian LASSO (BL) and BayesA respectively. Relative to the Gaussian density used in BGR, the DE and the scaled-t densities have higer mass at zero and thicker tails, inducing a different type of shrinkage. The DE and scaled-t prior densities can be represented as mixtures of scaled normal-densities of the form

$$p(\beta_{Rhj}|H) = \int N(\beta_{Rhj}|0, \sigma^2_{\beta hj}) p(\sigma^2_{\beta hj}|H) \partial \sigma^2_{\beta hj}$$

where $\sigma^2_{\beta hj}$ is a marker-specific variance parameter, $p(\sigma^2_{\beta hj}|H)$ is a prior density assigned to this variance parameter and H is a set of hyperparameters which may be specified a-priori or estimated from the data. When $p(\sigma^2_{\beta hj}|H)$ is an exponential (scaled-inverse chi-square) density, the resulting marginal prior density of marker effects is a double-exponential (scaled-t).

Pedigree-based regressions

They represent a generalization of the concept of 'family history' to complex genealogies. These regressions have been used over more than 5 decades for prediction of genetic values in animal and plant breeding applications. Pedigree regressions can be implemented by setting $\boldsymbol{K}_{hu} = \boldsymbol{A}$ where $\boldsymbol{A} = \{a(i,i')\}$ is a matrix whose entries are twice the coefficient of kinship between individuals, which can be computed from a pedigree.

Reproducing Kernel Hilbert Regressions(RKHS)

RKHS are used for semi-parametric regressions in applications as diverse as scatter-plot smoothing (smooting spline), spacial statistics (Kriging), gene expression or WGP. Estimates from RKHS can be motivated as the solution to a penalized optimization problem of as posterior modes in certain class of Bayesian models. A Bayesian formulation of RKHS can be implemented by simply setting $\boldsymbol{K}_{uh} = \{K_{uh}(i,i')\}$ to be a matrix whose entries contain the evaluations of a reproducing kernel at pairs of poings (i,i'). In WGP models the reproducing kernel, $K(i,i') = K(\boldsymbol{z}_i, \boldsymbol{z}'_i)$, maps from pairs of marker genotypes $(\boldsymbol{z}_i, \boldsymbol{z}'_i)$ onto co-variance function. For instance, using the Gaussian kernel, $K(i,i') = \exp\left(-\omega||\boldsymbol{z}_i - \boldsymbol{z}_{i'}||^2\right)$, where $||\boldsymbol{z}_i - \boldsymbol{z}_{i'}||$ is a Euclidean distance between the two vectors of marker genotypes and ω is a bandwith parameter.

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 b_i is unknown; therefore, the only information available is: $y_i < b_i$. In BGLR censored outcomes are then specified with three vectors, $\boldsymbol{y}$, $\boldsymbol{a}$ and $\boldsymbol{b}$. The configuration of the triplet for un-censored, right-censored, left-censored and interval censored are described in the table below.

	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

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

They can be 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 the type of 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 "Bernoulli". More details about this model can be found in Albert & Chib (1993).

Ordinal outcomes

They can be modeled using also the threshold model. Here we model, $\pi_{ij} = P(Y_i \leq j) = \Phi(\eta_{ij})$, where $\eta_{ij} = \gamma_j - \boldsymbol{x}_i'\boldsymbol{\beta}$, where $\Phi(\cdot)$ is the standard normal cumulative distribution function, γ_j is a threshold, the thresholds must satisfy, $-\infty = \gamma_0 < \gamma_1 < \cdots < \gamma_J = \infty$, J is the cardinality of $\boldsymbol{y}$. In order to run a regression for ordinal outomes, the response must be coded as $1, ..., J$, and the data should be ordered accordingly, the argument response_type should be set to "ordinal". More details about this model can be found in Albert & Chib (1993).

Value

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

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 $p(\lambda\vert\max,\alpha_1,\alpha_2) \propto Beta\left(\frac{\lambda}{\max}\vert\alpha_1,\alpha_2\right)$. 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,

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

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 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://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