Package 'BWGS'

July 30, 2020

Type Package

Title BreedWheat Genomic Selection Pipeline

Version 0.1.0

Description Package for Breed Wheat Genomic Selection Pipeline.
The R package 'BWGS' is developed by Gilles Charmet <gilles.charmet@inra.fr>.
This repository is forked from original repository <https://forgemia.inra.fr/umr-gdec/bwgs>
and modified as a R package.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports rrBLUP, BGLR, glmnet, randomForest, brnn, e1071

URL https://github.com/byzheng/BWGS

BugReports https://github.com/byzheng/BWGS/issues

RoxygenNote 7.1.0

NeedsCompilation no

Author Gilles Charmet [aut],
Louis Gautier Tran [aut],
Bangyou Zheng [cre]

Maintainer Bangyou Zheng <bangyou.zheng@csiro.au>

Repository CRAN

Date/Publication 2020-07-30 11:02:13 UTC

R topics documented:

bwgs.cv *Genomic Prediction with cross validation*

Description

The bwgs.cv function carries out cross-validation using genotypic and phenotypic data from a reference population, with options for genotypic matrix processing and genomic breeding value estimation.

Usage

```
bwgs.cv(
  geno,
  pheno,
  FIXED = "NULL",
  MAXNA = 0.2,
  MAF = 0.05,
  pop.reduct.method = "NULL",
  sample.pop.size = "NULL",
  geno.reduct.method = "NULL",
  reduct.marker.size = "NULL",
  pval = "NULL",
  r2 = "NULL",
  MAP = "NULL",
  geno.impute.method = "NULL",
  predict.method = "NULL",
  nFolds,
  nTimes
)
```

Arguments

geno	Matrix (n x m) of genotypes for the training population: n lines with m markers. Genotypes should be coded -1, 0, 1. Missing data are allowed and coded as NA.
pheno	Vector (n x 1) of "phenotypes", i.e. observations or pre-processed, corrected values. This vector should have no missing values, otherwise missing values (NA) will be omitted in both pheno and geno. In a first step, bwgs.cv checks whether rownames(geno) match with names(pheno). If not the case, the common elements (intersect) are selected in both geno and pheno for further analyses. If a MAP file is provided, the selected set of markers are also sorted out in MAP.
FIXED	A matrix of fixed effect, to be used with some methods such as those included in BGLR, MUST have same rownames as geno and coded(-1 0 1)
MAXNA	The maximum proportion of missing value which is admitted for filtering marker columns in geno. Default value is 0.2
MAF	The minimum allele frequency for filtering marker colums in geno;default value is 0.05

`pop.reduct.method`

Method for reducing the size of the training population. Can be used for teaching purposes, no real interest in real life if the entire population is already genotyped and phenotyped. Default value is NULL (all training set used). Proposed methods are:

- RANDOM: a subset of sample.pop.size is randomly selected for training the model, and the unselected part of the population is used for validation. The process is repeated nFolds * nTimes to have the same number of replicates than with cross-validation.
- OPTI: the optimization algorithm based on CDmean (Rincent et al 2012) to select a subset which maximizes average CD (coefficient of determination) in the validation set. Since the process is long and has some stochastic component, it is repeated only nTimes.

`sample.pop.size`

The size of the subset of individuals in the training set (both geno and pheno) selected by pop.reduct.method if not NULL.

`geno.reduct.method`

Allows sampling a subset of markers for speeding up computing time and/or avoid introducing more noise than informative markers. Options are:

- RMR: Random sampling (without replacement) of a subset of markers. To be used with the parameter "reduct.marker.size".
- LD (with r2 and MAP): enables "pruning" of markers which are in LD > r2. Only the marker with the least missing values is kept for each pair in LD>r2. To allow faster computation, r2 is estimated chromosome by chromosome, so a MAP file is required with information of marker assignation to chromosomes. The MAP file should contain at least three columns: marker_name, chromosome_name and distance_from_origin (either genetic of physical distance, only used for sorting markers, LD being re-estimated from marker Data).
- ANO (with pval): one-way ANOVA are carried out with R function lm on trait "pheno". Every markers are tested one at a time, and only markers with pvalue<pval are kept for GEBV prediction.
- ANO+LD (with pval and r2, MAP is facultative): combines a first step of marker selection with ANO, then a second step of pruning using LD option.

`reduct.marker.size`

Specifies the number of markers for the genotypic reduction using RMR (reduct.size < m).

`pval` p value for ANO method, 0 < pval < 1.

`r2` Coefficient of linkage disequilibrium (LD). Setting 0<r2<1 if the genotypic reduction method is in LD or ANO+LD .

`MAP` A matrix with markers in rows and at least ONE columns with colnames= "chrom". Used for computing r2 within linkage groups.

`geno.impute.method`

Allow missing marker data imputation using the two methods proposed in function A.mat of package rrBLUP, namely:

- MNI: missing data are replaced by the mean allele frequency of the marker (column in geno)
- EMI: missing data are replaced using an expectation-maximization methods described in function A.mat (Endelman & Janninck 2012).

Default value is NULL.

Note that these imputation methods are only suited when there are a few missing value, typically in marker data from SNP chips of KasPAR. They are NOT suited for imputing marker data from low density to high density designs, and when there are MANY missing Data as typically provided by GBS. More sophisticated software (e.g. Beagles, Browning & Browning 2016) should be used before BWGS.

`predict.method` The options for genomic breeding value prediction methods. The available options are:

- GBLUP: performs G-BLUP using a marker-based relationship matrix, implemented through BGLR R-library. Equivalent to ridge regression (RR-BLUP) of marker effects.
- EGBLUP: performs EG-BLUP, i.e. BLUP using a "squared" relationship matrix to model epistatic 2x2 interactions, as described by Jiang & Reif (2015), using BGLR library
- RR: ridge regression, using package glmnet. In theory, strictly equivalent to gblup.
- LASSO: Least Absolute Shrinkage and Selection Operator is another penalized regression methods which yield more shrinked estimates than RR. Run by glmnet library.
- EN: Elastic Net (Zou and Hastie, 2005), which is a weighted combination of RR and LASSO, using glmnet library

Several Bayesian methods, using the BGLR library:

- BRR: Bayesian ridge regression: same as rr-blup, but bayesian resolution. Induces homogeneous shrinkage of all markers effects towards zero with Gaussian distribution (de los Campos et al, 2013)
- BL: Bayesian LASSO: uses an exponential prior on marker variances priors, leading to double exponential distribution of marker effects (Park & Casella 2008)
- BA: Bayes A uses a scaled-t prior distribution of marker effects. (Meuwissen et al 2001).
- BB: Bayes B, uses a mixture of distribution with a point mass at zero and with a slab of non-zero marker effects with a scaled-t distribution (Habier et al 2011).
- BC: Bayes C same as Bayes B with a slab with Gaussian distribution.

A more detailed description of these methods can be found in Perez & de los Campos 2014 (http://genomics.cimmyt.org/BGLR-extdoc.pdf). Three semi-parametric methods:

- RKHS: reproductive kernel Hilbert space and multiple kernel MRKHS, using BGLR (Gianola and van Kaam 2008). Based on genetic distance and a kernel function to regulate the distribution of marker effects. This methods is claimed to be effective for detecting non additive effects.

- RF: Random forest regression, using randomForest library (Breiman, 2001, Breiman and Cutler 2013). This methods uses regression models on tree nodes which are rooted in bootstrapping data. Supposed to be able to capture interactions between markers
- SVM: support vector machine, run by e1071 library. For details, see Chang, Chih-Chung and Lin, Chih-Jen: LIBSVM: a library for Support Vector Machines http://www.csie.ntu.edu.tw/~cjlin/libsvm
- BRNN: Bayesian Regularization for feed-forward Neural Network, with the R-package BRNN (Gianola et al 2011). To keep computing time in reasonable limits, the parameters for the brnn function are neurons=2 and epochs = 20.

`nFolds` Number of folds for the cross-validation. Smallest value recommended is nFolds = 3.

`nTimes` Number of independent replicates for the cross-validation. Smallest value recommended is nTimes = 3.

Value

The class bwgs.cv returns a list containing:

- **summary**: Summary of cross-validation, including mean and standard deviation of predictive ability (i.e. correlation between phenotype and GEBV, estimated on the validation fold, then averaged over replicates (nTimes), Time taken by the computation and number of markers
- **cv**: Vector of predictive abilities averaged over nFolds, for each of the nTimes replicates
- **sd**: Standard deviation of the nTimes predictive abilities
- **MSEP**: Square root of the mean-squared error of prediction, averaged over Ntimes
- **SDMSEP**: Standard deviation of the Square root of the mean-squared error of prediction, averaged over Ntimes
- **bv_table**: Matrix of dimension n x 4. Columns are:
  - Real BV, i.e. pheno vector
  - Predict BV: the nx1 vector of GEBVs
  - gpreSD: Standart deviation of estimated GEBV
  - CD: coefficient of determination for each GEBV, estimated as sqrt Note that gpredSD and CD are only available for methods using the BGLR library, namely GBLUP, EGBLUP, BA,BB,BC,BL,RKHS and MKRKHS. These two columns contain NA for methods RF, RR, LASSO, EN and SVM.

Examples

```
data(inra)
# Cross validation using GBLUP method
cv_gblup <- bwgs.cv(TRAIN47K, YieldBLUE,
                      geno.impute.method = "mni",
                      predict.method = "gblup",
                      nFolds = 10,
                      nTimes = 1)
```

bwgs.predict *Computes the GEBV prediction for the target population with only genotypic Data using the options for model selection.*

Description

Computes the GEBV prediction for the target population with only genotypic Data using the options for model selection.

Usage

```
bwgs.predict(
  geno_train,
  pheno_train,
  geno_target,
  FIXED_train = "NULL",
  FIXED_target = "NULL",
  MAXNA = 0.2,
  MAF = 0.05,
  geno.reduct.method = "NULL",
  reduct.size = "NULL",
  r2 = "NULL",
  pval = "NULL",
  MAP = "NULL",
  geno.impute.method = "NULL",
  predict.method = "GBLUP"
)
```

Arguments

geno_train	Matrix (n x m) of genotypes for the training population: n lines with m markers. Genotypes should be coded as -1, 0, 1, NA. Missing data are allowed and coded as NA.
pheno_train	Vector (n x 1) of phenotype for the training phenotypes. This vector should have no missing values. Otherwise, missing values (NA) will be omitted in both pheno_train and geno_train.
geno_target	Matrix (z x m) of genotypes for the target population: z lines with the same m markers as in geno_train. Genotypes should be coded as -1, 0, 1, NA. Missing data are allowed and coded as NA. Other arguments are identical to those of bwgs.cv, except pop_reduct_method, nTimes and nFolds, since the prediction is run only once, using the whole training population for model estimation, then applied to the target population.
FIXED_train	A matrix of fixed effect for training, to be used with some methods such as those included in BGLR, MUST have same rownames as geno and coded(-1 0 1)

`FIXED_target` A matrix of fixed effect for targeting, to be used with some methods such as those included in BGLR, MUST have same rownames as geno and coded(-1 0 1)

`MAXNA` The maximum proportion of missing value which is admitted for filtering marker columns in geno. Default value is 0.2

`MAF` The minimum allele frequency for filtering marker colums in geno; default value is 0.05

`geno.reduct.method`

Allows sampling a subset of markers for speeding up computing time and/or avoid introducing more noise than informative markers. Options are:

- RMR: Random sampling (without replacement) of a subset of markers. To be used with the parameter "reduct.marker.size".
- LD (with r2 and MAP): enables "pruning" of markers which are in LD > r2. Only the marker with the least missing values is kept for each pair in LD>r2. To allow faster computation, r2 is estimated chromosome by chromosome, so a MAP file is required with information of marker assignation to chromosomes. The MAP file should contain at least three columns: marker_name, chromosome_name and distance_from_origin (either genetic of physical distance, only used for sorting markers, LD being re-estimated from marker Data).
- ANO (with pval): one-way ANOVA are carried out with R function lm on trait "pheno". Every markers are tested one at a time, and only markers with pvalue<pval are kept for GEBV prediction.
- ANO+LD (with pval and r2, MAP is facultative): combines a first step of marker selection with ANO, then a second step of pruning using LD option.

`reduct.size` Specifies the number of markers for the genotypic reduction using RMR (reduct.size < m).

`r2` Coefficient of linkage disequilibrium (LD). Setting 0<r2<1 if the genotypic reduction method is in LD or ANO+LD .

`pval` p value for ANO method, 0 < pval < 1.

`MAP` A file with markers in rows ane at least ONE columns with colnames= "chrom". Used for computing r2 within linkage groups.

`geno.impute.method`

Allow missing marker data imputation using the two methods proposed in function A.mat of package rrBLUP, namely:

- MNI: missing data are replaced by the mean allele frequency of the marker (column in geno)
- EMI: missing data are replaced using an expectation-maximization methods described in function A.mat (Endelman & Janninck 2012).

Default value is NULL.

Note that these imputation methods are only suited when there are a few missing value, typically in marker data from SNP chips of KasPAR. They are NOT suited for imputing marker data from low density to high density designs, and when there are MANY missing Data as typically provided by GBS. More sophisticated software (e.g. Beagles, Browning & Browning 2016) should be used before BWGS.

`predict.method` The options for genomic breeding value prediction methods. The available options are:

- GBLUP: performs G-BLUP using a marker-based relationship matrix, implemented through BGLR R-library. Equivalent to ridge regression (RR-BLUP) of marker effects.
- EGBLUP: performs EG-BLUP, i.e. BLUP using a "squared" relationship matrix to model epistatic 2x2 interactions, as described by Jiang & Reif (2015), using BGLR library
- RR: ridge regression, using package glmnet. In theory, strictly equivalent to gblup.
- LASSO: Least Absolute Shrinkage and Selection Operator is another penalized regression methods which yield more shrinked estimates than RR. Run by glmnet library.
- EN: Elastic Net (Zou and Hastie, 2005), which is a weighted combination of RR and LASSO, using glmnet library

Several Bayesian methods, using the BGLR library:

- BRR: Bayesian ridge regression: same as rr-blup, but bayesian resolution. Induces homogeneous shrinkage of all markers effects towards zero with Gaussian distribution (de los Campos et al, 2013)
- BL: Bayesian LASSO: uses an exponential prior on marker variances priors, leading to double exponential distribution of marker effects (Park & Casella 2008)
- BA: Bayes A uses a scaled-t prior distribution of marker effects. (Meuwissen et al 2001).
- BB: Bayes B, uses a mixture of distribution with a point mass at zero and with a slab of non-zero marker effects with a scaled-t distribution (Habier et al 2011).
- BC: Bayes C same as Bayes B with a slab with Gaussian distribution.

A more detailed description of these methods can be found in Perez & de los Campos 2014 (http://genomics.cimmyt.org/BGLR-extdoc.pdf). Three semi-parametric methods:

- RKHS: reproductive kernel Hilbert space and multiple kernel MRKHS, using BGLR (Gianola and van Kaam 2008). Based on genetic distance and a kernel function to regulate the distribution of marker effects. This methods is claimed to be effective for detecting non additive effects.
- RF: Random forest regression, using randomForest library (Breiman, 2001, Breiman and Cutler 2013). This methods uses regression models on tree nodes which are rooted in bootstrapping data. Supposed to be able to capture interactions between markers
- SVM: support vector machine, run by e1071 library. For details, see Chang, Chih-Chung and Lin, Chih-Jen: LIBSVM: a library for Support Vector Machines http://www.csie.ntu.edu.tw/~cjlin/libsvm
- BRNN: Bayesian Regularization for feed-forward Neural Network, with the R-package BRNN (Gianola et al 2011). To keep computing time in reasonable limits, the parameters for the brnn function are neurons=2 and epochs = 20.

Value

The object bwgs.predict returns Matrix of dimension nx3. Columns are:

- Predict BV: the nx1 vector of GEBVs for the validation set (rows of geno_valid)
- gpredSD: Standart deviation of estimated GEBV
- CD: coefficient of determination for each GEBV, estimated as sqrt ((1-stdev(GEBVi))^2/2g)

Note that gpredSD and CD are only available for methods using the BGLR library, namely GBLUP, EGBLUP, BA,BB,BC,BL,RKHS and MKRKHS. These two columns contain NA for methods RF, RR, LASSO, EN and SVM.

Examples

```
data(inra)
# Prediction using GBLUP method
predict_gblup <- bwgs.predict(geno_train = TRAIN47K,
     pheno_train = YieldBLUE,
     geno_target = TARGET47K,
     MAXNA = 0.2,
     MAF = 0.05,
     geno.reduct.method = "NULL",
     reduct.size = "NULL",
     r2 = "NULL",
     pval = "NULL",
     MAP = "NULL",
     geno.impute.method = "MNI",
     predict.method = "GBLUP")
```

---

inra47k *INRA47K*

Description

inra data contains a set of geno47K(760 x 47839), pheno (760 x 1) and MAP47K (47839 x 3). The phenotype pheno contains adjusted genotype means for yield trait (YLD) over multi-year/location trials.

Usage

```
MAP47K

TARGET47K

TRAIN47K

YieldBLUE
```

Format

An object of class `data.frame` with 10000 rows and 3 columns.

An object of class `matrix` (inherits from `array`) with 100 rows and 10000 columns.

An object of class `matrix` (inherits from `array`) with 100 rows and 10000 columns.

An object of class `numeric` of length 100.

Source

<https://forgemia.inra.fr/umr-gdec/bwgs>

Index