Package 'G2P'

August 15, 2017

Title Genomic Selection Prediction and Evalution

Version 1.1.0

Description This package provide a platform for Genomic Selection.Researchers could apply this package to GS modeling as well as Prediction and results evalution.

Depends R (>= 3.2.0)

Imports BGLR (>= 1.0.5), PRROC(>= 1.1), e1071(>= 1.6-7), glmnet(>= 2.0), randomForest(>= 4.6-12), rrBLUP(>= 4.4), snowfall(>= 1.84-6.1), spls(>= 2.2-1), brnn(>= 0.6), sommer(>= 2.6), hglm(>= 2.1-1), hglm.data(>= 1.0-0)

License GPL-2|GPL-3

Suggests rgl(>= 0.97.0), pheatmap(>= 1.0.8), impute(>= 1.46.0)

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Encoding UTF-8

Author Chuang Ma [aut, cre], Qian Cheng[cre], Zhixu Qiu , Jie Song

Maintainer Chuang Ma <chuangma2006@gmail.com>, Qian Cheng<qchengray@gmail.com>

R topics documented:

`cvSampleIndex` *Generate Sample Indices for Training Sets and Testing Sets*

Description

This function generates indices for samples in training and testing sets for performing the N-fold cross validation experiment.

Usage

```
cvSampleIndex(sampleNum, cross = 5, seed = 1, randomSeed = FALSE)
```

Arguments

`sampleNum`	The number of samples needed to be partitioned into training and testing sets.
`cross`	The fold of cross validation.
`seed`	An integer used as the seed for data partition. The default value is 1.
`randomSeed`	Logical variable, default FALSE.

Value

A list and each element including $trainIdx $testIdx and $cvIdx

$trainIdx The index of training samples.

$testIdx The index of testing samples.

$cvIdx The cross validation index.

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

Examples

```
## Not run:
## Load example data ##
data(GYSS)
## leave-one out cross validation
a <- cvSampleIndex(sampleNum = nrow(Markers), cross = nrow(Markers), seed = 1)

## 5-fold cross validation
b <- cvSampleIndex(sampleNum = nrow(Markers), cross = 5, seed = 1)

## End(Not run)
```

evaluateGS *evaluateGS*

Description

this function is used to evaluete the accuracy of predicted by genomic selection model.

Usage

```
evaluateGS(realScores, predScores, Probability = FALSE, evalMethod = "RE",
  Beta = 1, BestIndividuals = "top", topAlpha = 1:90, allNDCG = F,
  globalAlpha = F, probIndex = NULL)
```

Arguments

realScores	(numeric,vector)vector is the real breeding values of the validation individual for a trait.
predScores	(numeric,vector or matrix)the prediction breeding value predicted by genomic selection model of the individuals.
Probability	(logical)whether the predScores is probability? Default FALSE.
evalMethod	(character)the evaluation methods selected to evaluate, which include "pearson", "kendall", "spearman", "MSE","R2" "RE", "Kappa", "AUC","AUCpr","accuracy","F1","meanNDCG", "NDCG".
Beta	(numeric)the parameter of "F1".
BestIndividuals	(character)the position of expected phenotype in whole phenotypic data set."top","buttom" or "middle",default "top".
topAlpha	(numeric,(0,100])a vector is the proportion of excellent individuals,default 1:90.
globalAlpha	(logical)indicates if evaluate global methods(pearson, kendall, spearman, MSE and R2) by alpha,default FALSE.
probIndex	(integer)indicates the column index which prediction result is probability.

Value

a list inculding evaluation results with methods which user selected.

Author(s)

Chuang Ma, Zhixu Qiu, Qian Cheng, Jie Song

Examples

```
## Not run:
data(GYSS)
########## predicting breeding value
predlist <-  G2PCrossValidation(cross = 10,seed = 1 ,cpus = 3,markers  = Markers,pheVal  = phenotype,
                 modelMethods = c("rrBLUP","RFC"),outputModel = FALSE)
predMartix <- NULL
for(ii in 1:10){predMartix <- rbind(predMartix,predlist[[ii]])}
######## evaluate the accuracy of the prediction result

evaluareTest <- evaluateGS(realScores = predMartix[,1], predScores = predMartix[,2:3],
                           evalMethod = c("pearson", "kendall","spearman","RE","Kappa",
                                         "AUC","AUCpr","NDCG","meanNDCG",
                                 "MSE","R2","F1","accuracy"),topAlpha = 1:90, probIndex = 2)

## End(Not run)
```

---

feature_assess *Feature Selection*

Description

This function scores each marker,so that reduce the data dimension and perform feature selection.(Methods including Gini,Accuracy and rrBLUP).

Usage

```
feature_assess(markers, phenotype, method = c("rrBLUP", "Gini", "Accuracy"),
  ntree = 500, importance = TRUE, posPercentage = 0.4,
  BestIndividuals = c("top", "middle", "buttom"))
```

Arguments

`markers` (numeric, matrix)row is sample well column is SNP information (feature).Genotypes should be coded as 0,1,2;0 represent AA(homozygote),2 represent BB(homozygote) and 1 represent AB(heterozygote);missing (NA) alleles are not allowed.

`phenotype` (numeric)the phenotype value of each individual.

`method` (character)the method of feature selction including "Gini" "Accuracy" "rrBLUP", default "RR-BLUP" `ntree` (numeric)the number of random forest decision tree, default 500

`importance` (logical)whether the results of variable importance,default TRUE

`posPercentage` (numeric,[0,1])phenotype of extreme individuals which expected, default 0.4

`BestIndividuals`
(character)the position of expected phenotype in whole phenotypic data set."top","buttom" or "middle",defult "top".

Value

A numeric mode score of each position of SNPs

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

Examples

```
## Not run:
data(GYSS)
## feature selection with Gini ##
Gini_selection <- feature_assess(markers = Markers, phenotype = phenotype, method = "Gini",
                                 ntree = 500, importance = TRUE, posPercentage = 0.40,
                                 BestIndividuals = "top")

## feature selection with Acc ##
Acc_selection <- feature_assess(markers = Markers, phenotype = phenotype, method = "Accuracy",
                                ntree = 500, importance = TRUE, posPercentage = 0.40,
                                BestIndividuals = "top")

## feature selection with rrBLUP ##
rrBLUP_selection <- feature_assess(markers = Markers, phenotype = phenotype, method = "rrBLUP",
                                   posPercentage = 0.40, BestIndividuals = "top")

## End(Not run)
```

fit.BGLR *Fit Regression Model*

Description

This function can fit several regression models of genomic selection such as BayesA, BayesB, BayesC, BRR(BayesBayesian Ridge Regression) and BL(Bayesian LASSO).

Usage

```
fit.BGLR(trainedMarkerMat, trainedPheVal, predictMarkerMat, modelMethods,
  outputModel = FALSE, nIter = 1500, burnIn = 500, thin = 5,
  saveAt = "", S0 = NULL, df0 = 5, R2 = 0.5, weights = NULL,
  verbose = FALSE, rmExistingFiles = TRUE, groups = NULL)
```

Arguments

`trainedMarkerMat`
(numeric, matrix)each row is the each training sets individual's SNP genotypes informations.Genotypes should be coded as 0,1,2;0 represent AA(homozygote),2 represent BB(homozygote) and 1 represent AB(heterozygote); missing (NA) alleles are not allowed.

`predictMarkerMat`
(numeric, matrix)each row is the each testing sets individual's SNP genotypes informations.Genotypes should be coded as 0,1,2;0 represent AA(homozygote),2 represent BB(homozygote) and 1 represent AB(heterozygote); missing (NA) alleles are not allowed.

`modelMethods` (character)the model to fit. "BayesA", "BayesB", "BayesC", "BL", "BRR".

`outputModel` (logical)if TRUE, return the list of training model and prediction result, default FALSE.

`nIter, burnIn, thin`
(integer)the number of iterations, burn-in and thinning,default nIter 7000,burnIn 500,thin 5.

`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,default "".

`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.Default S0 NULL,df0 5.

`R2` (numeric, (0,1))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-extdoc.pdf.Defult 0.5

`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.

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

`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.

`trainedphenotype`
(numeric)the phenotype Value of each individual.

Value

a list including model and prediction result (outputModel = TRUE) a array of prediction result (outputModel = FALSE)

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

See Also

BGLR

Examples

```
## Not run:
## Load example data ##
data(GYSS)

## Fit Bayes A model and prediction ##
BayesA_model <- fit.BGLR(trainedMarkerMat = Markers[1:200,], predictMarkerMat = Markers[201:242,],
                    trainedPheVal = phenotype[1:200], modelMethods = "BayesA", outputModel = TRUE)

## End(Not run)
```

`fit.BRNN` *Modeling and predicting using Bayesian Regularization Neural Networks(BRNN)*

Description

This function can fit BRNN model and export the prediction value of testing sets.

Usage

```
fit.BRNN(trainedMarkerMat, trainedPheVal, predictMarkerMat,
  outputModel = FALSE, verbose = TRUE, neurons = 4, epochs = 30,
  cpus = 1, ...)
```

Arguments

`trainedMarkerMat`
: (numeric)a matrix, each row is the each training sets individual's SNP genotypes informations. Genotypes should be coded as 0,1,2; 0 represent AA(homozygote),2 represent BB(homozygote) and 1 represent AB(heterozygote);missing (NA) alleles are not allowed.

`trainedPheVal`
: (numeric)the phenotype Value of each individual.

`predictMarkerMat`
: (numeric)a matrix, each row is the each testing sets individual's SNP genotypes informations. Genotypes should be coded as 0,1,2; 0 represent AA(homozygote),2 represent BB(homozygote) and 1 represent AB(heterozygote);missing (NA) alleles are not allowed.

`outputModel`
: (logical)if true, return the list of training model.

`verbose` (logical)if TRUE, will print iteration history.

`neurons` (integer)indicates the number of neurons,defult 4.

`epochs` (integer)maximum number of epochs(iterations) to train, default 30.

`cpus` (integer)number of cpu cores to be used for calculations (only available in UNIX-like operating systems), default 1.

`...` other parameters, details see package brnn

Value

a list including model and prediction result (outputModel = TRUE) a array of prediction result (outputModel = FALSE)

Author(s)

Chuang Ma , Qian Cheng , Zhixu Qiu , Jie Song

See Also

brnn

Examples

```
## Not run:
## Load example data ##
data(GYSS)

## use RR model to modeling and predict ##
BRNN_Res <- fit.BRNN(trainedMarkerMat = Markers,trainedPheVal = phenotype,
                     predictMarkerMat = Markers[1:10,],cpus = 1 )

## End(Not run)
```

`fit.mmer` *Modeling and predicting using methods AI, NR , EM , EMMA.*

Description

This function can fit AI, NR , EM , EMMA model and export the prediction values of testing sets.

Usage

```
fit.mmer(trainedMarkerMat, trainedPheVal, predictMarkerMat, Z = NULL,
  X = NULL, method = "NR", effectConcider = "A", outputModel = FALSE,
  iters = 20, cpus = 1, REML = TRUE, ...)
```

Arguments

`trainedMarkerMat`
(numeric)a matrix, each row is the each training sets individual's SNP genotypes informations. Genotypes should be coded as 0,1,2; 0 represent AA(homozygote), 2 represent BB(homozygote) and 1 represent AB(heterozygote); missing (NA) alleles are not allowed.

`trainedPheVal` (numeric)the phenotype Value of each individual.

`predictMarkerMat`
(numeric)a matrix, each row is the each testing sets individual's SNP genotypes informations. Genotypes should be coded as 0,1,2; 0 represent AA(homozygote), 2 represent BB(homozygote) and 1 represent AB(heterozygote); missing (NA) alleles are not allowed.

`Z` A 2-level list,incidence matrices and var-cov matrices for random effects. This works for ONE OR MORE random effects. This needs to be provided as a 2-level list structure, defult NULL.

`X` (numeric, matrix) design matrix related to the parameters not to be shrunk (i.e. fixed effects in the mixed model framework),defult no shrink.

`method` (string)select the algorithm to fit the model including NR, AI, EM, EMMA.

`effectConcider` (string)if Z = NULL, random effects are auto generated.

`outputModel` (logical)if true, return the list of training model.

`iters` (numeric)a scalar value indicating how many iterations have to be performed if the optimization algorithms. There is no rule of tumb for the number of iterations but less than 8 is usually enough, default 20.

`cpus` (numeric)number of cores to use when the user passes multiple responses in the model for a faster execution of univariate models. The default is 1

`REML` (logical)indicating if restricted maximum likelihood should be used instead of ML. The default is TRUE.

`...` other parameters, details see package sommer.

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

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

Value

a list including model and prediction result (outputModel = TRUE) a array of prediction result (outputModel = FALSE)

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

See Also

sommer

Examples

```
## Not run:
## Load example data ##
data(GYSS)

## use RR model to modeling and predict ##
EM_Res <- fit.mmer(trainedMarkerMat = Markers[1:200,], trainedPheVal = phenotype,
                   predictMarkerMat = Markers[201:242,], method = "EM",
                   effectConcider = "A", iters = 20,
                   )

## End(Not run)
```

fit.RKHS *Modeling and predicting using Reproducing Kernel Hilbert Space(RKHS).*

Description

This function can fit RKHS model and export the prediction value of testing sets.

Usage

```
fit.RKHS(trainedMarkerMat, trainedPheVal, predictMarkerMat,
  outputModel = FALSE, nIter = 1500, burnIn = 500, thin = 5,
  saveAt = "", S0 = NULL, df0 = 5, R2 = 0.5, weights = NULL,
  verbose = FALSE, rmExistingFiles = TRUE, groups = NULL)
```

Arguments

`trainedMarkerMat`
: (numeric)a matrix, each row is the each training sets individual's SNP genotypes informations. Genotypes should be coded as 0,1,2; 0 represent AA(homozygote), 2 represent BB(homozygote) and 1 represent AB(heterozygote); missing (NA) alleles are not allowed.

`trainedPheVal`
: (numeric)the phenotype Value of each individual.

`predictMarkerMat`
: (numeric)a matrix, each row is the each testing sets individual's SNP genotypes informations. Genotypes should be coded as 0,1,2; 0 represent AA(homozygote), 2 represent BB(homozygote) and 1 represent AB(heterozygote); missing (NA) alleles are not allowed.

`outputModel`
: (logical)if true, return the list of training model.

`nIter, burnIn, thin`
: (integer)the number of iterations, burn-in and thinning,default nIter 1500, burnIn 500, thin 5.

`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,default "".

`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. Default S0 NULL,df0 5.

`R2` (numeric, (0,1))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-extdoc.pdf. Defult 0.5

`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.

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

`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.

Value

a list including model and prediction result (outputModel = TRUE) a array of prediction result (outputModel = FALSE)

Author(s)

Chuang Ma , Qian Cheng , Zhixu Qiu , Jie Song

See Also

BGLR

Examples

```
## Not run:
## Load example data ##
data(GYSS)

## use RR model to modeling and predict ##
RKHS_Res <- fit.RKHS(trainedMarkerMat = Markers, trainedPheVal = phenotype,
                      predictMarkerMat = Markers[1:10,],nIter = 1500, burnIn = 500)

## End(Not run)
```

fit.RR *Modeling and Predicting using ridge regression*

Description

This function can fit ridge regression model and export the prediction value of testing sets.

Usage

```
fit.RR(trainedMarkerMat, trainedPheVal, predictMarkerMat, cpus = 1,
  outputModel = FALSE)
```

Arguments

trainedMarkerMat	(numeric,matrix)each row is the each training sets individual's SNP genotypes informations. Genotypes should be coded as 0,1,2; 0 represent AA(homozygote), 2 represent BB(homozygote) and 1 represent AB(heterozygote); missing (NA) alleles are not allowed.
trainedPheVal	(numeric)the phenotype Value of each individual.
predictMarkerMat	(numeric,matrix)each row is the each testing sets individual's SNP genotypes informations. Genotypes should be coded as 0,1,2; 0 represent AA(homozygote), 2 represent BB(homozygote) and 1 represent AB(heterozygote); missing (NA) alleles are not allowed.
cpus	(integer)number of cpu cores to use for calculations (only available in UNIX-like operating systems), default 1.
outputModel	(logical)if true, return the list of training model.

Value

a list including model and prediction result (outputModel = TRUE) a array of prediction result (outputModel = FALSE)

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

See Also

rrBLUP

Examples

```
## Not run:
## Not run:
## Load example data ##
data(GYSS)

## use RR model to modeling and predict ##
rr_Res <- fit.RR(trainedMarkerMat = Markers, trainedPheVal = phenotype,
                 predictMarkerMat = Markers[1:10,], cpus = 1 )

## End(Not run)
```

G2P *Genotype to Phenotype*

Description

this function is to predict phenotype from genotype.

Usage

```
G2P(trainMarker, trainPheno, testMarker, testPheno = NULL,
  modelMethods = "BayesA", outputModel = FALSE, nIter = 1500,
  burnIn = 500, thin = 5, saveAt = "", S0 = NULL, df0 = 5, R2 = 0.5,
  weights = NULL, verbose = FALSE, rmExistingFiles = TRUE,
  groups = NULL, importance = FALSE, posPercentage = 0.4,
  BestIndividuals = c("top"), ntree = 500, nodesize = 1,
  kernel = c("linear"), gamma = 1, cost = 2^(-9), K = 8, eta = 0.7,
  select = "pls2", fit = "simpls", scale.x = FALSE, scale.y = FALSE,
  eps = 1e-04, trace = FALSE, maxstep = 100, alpha = 1, X = NULL,
  family = gaussian(link = identity), lambda = NULL, tol.err = 1e-06,
  tol.conv = 1e-08, epochs = 30, neurons = 4, Z = NULL,
  effectConcider = "A", mmerIters = 20, ...)
```

Arguments

`trainMarker`	(numeric,matrix)each row is the each training sets individual's SNP genotypes informations. Genotypes should be coded as 0,1,2; 0 represent AA(homozygote), 2 represent BB(homozygote) and 1 represent AB(heterozygote); missing (NA) alleles are not allowed.
`trainPheno`	(numeric)the phenotype value of each individual.
`testMarker`	(numeric,matrix)each row is the each testing sets individual's SNP genotypes informations. Genotypes should be coded as 0,1,2; 0 represent AA(homozygote), 2 represent BB(homozygote) and 1 represent AB(heterozygote); missing (NA) alleles are not allowed.
`testPheno`	(numeric)the phenotype value of test population individual, default NULL.

`modelMethods` the model to fit."BayesA", "BayesB", "BayesC", "BL", "BRR","RKHS","rrBLUP","LASSO","SPLS","SV "RR", "RKHS", "BRNN", "EM", "EMMA", "AI", "NR".

`outputModel` (logical)if true, return the list of training model.

`nIter, burnIn, thin`

(integer) the number of iterations, burn-in and thinning,default nIter 7000,burnIn 500,thin 5.

`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,default ""

`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.Default S0 NULL,df0 5.

`R2` (numeric, (0,1)) 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-extdoc.pdf.Defult 0.5

`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.

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

`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.

`importance` RandomForest parameter:Should importance of predictors be assessed?Defualt FALSE

`posPercentage` (numeric)the percentage positive samples in all samples.1 > posPercentage > 0.

`BestIndividuals`

It is a position that the best individuals (positive samples) in a training group, according to the breeding values of a training group's trait. if the trait was yield,flowering or disease resistance,and male flowering time to female flowering time,it is "top"(default), "buttom",and "middle" of the breeding values, respectively.

`ntree` RandomForest parameter:Number of trees to grow. This should not be set to too small a number, to ensure that every input row gets predicted at least a few times.Defualt 500

`nodesize` Randomforest parameter Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). Note that the default values are different for classification (1) and regression (5).

`kernel` svm parameter the kernel used in training and predicting. You might consider changing some of the following parameters, depending on the kernel type.(linear,polynomial,sigmoid,radi "linear".

`gamma` svm parameter parameter needed for all kernels except linear (default: 1/(data dimension))

`cost` svm cost,default 2^(-9)

`K` Number of hidden components

`eta` Thresholding parameter. eta should be between 0 and 1.

`select` PLS algorithm for variable selection. Alternatives are "pls2" or "simpls". Default is "pls2"

`fit` PLS algorithm for model fitting. Alternatives are "kernelpls", "widekernelpls", "simpls", or "oscorespls". Default is "simpls".

`scale.x` Scale predictors by dividing each predictor variable by its sample standard deviation?

`scale.y` Scale responses by dividing each response variable by its sample standard deviation?

`eps` An effective zero. Default is 1e-4

`trace` Print out the progress of variable selection?

`maxstep` Maximum number of iterations when fitting direction vectors. Default is 100.

`alpha` The elasticnet mixing parameter.Detail in glmnet.

`X` (numeric, matrix) design matrix related to the parameters not to be shrunk (i.e. fixed effects in the mixed model framework),defult no shrink.

`family` the distribution family of y, see help('family') for more details.

`lambda` the shrinkage parameter determines the amount of shrinkage. Default is NULL meaning that it is to be estimated along with other model parameters.

`tol.err` internal tolerance level for extremely small values; default value is 1e-6.

`tol.conv` tolerance level in convergence; default value is 1e-8.

`epochs` (integer)maximum number of epochs(iterations) to train, default 30.

`neurons` (integer)indicates the number of neurons,defult 4.

`Z` (2-level list)incidence matrices and var-cov matrices for random effects. This works for ONE OR MORE random effects. This needs to be provided as a 2-level list structure, defult NULL.

`effectConcider` (string)if Z = NULL, random effects are auto generated.

`...` arguments passed to or from other methods.

`iters` (numeric)a scalar value indicating how many iterations have to be performed if the optimization algorithms. There is no rule of tumb for the number of iterations but less than 8 is usually enough, default 20.

Value

a matrix: The prediction results of multi-methods

Author(s)

Chuang Ma ,Qian Cheng, Zhixu Qiu,Jie Song

See Also

GSmachine GSReModel fit.RR fit.BRNN fit.RKHS fit.mmer predictGS

Examples

```
## Not run:
data(GYSS)
########## predicting breeding value
predRes <- G2P(Markers[1:200,],phenotype[1:200],Markers[201:242,],
                phenotype[201:242],modelMethods = c("rrBLUP", "RFC"),
                outputModel = FALSE)

## End(Not run)
```

G2P.app *Genotypes to phenotypes with Shiny App*

Usage

```
G2P.app()
```

Value

the Shiny App.

See Also

`shiny`

Examples

```
TSIS.app()
```

`G2PCrossValidation` *G2PCrossValidation*

Description

this function is apply cross validation to test Genomic Selection model trained by different methods and datas.

this function is apply cross validation to test Genomic Selection model trained by different methods and datas.

Usage

```
G2PCrossValidation(cross = 5, seed = 1, cpus = 1, markers, pheVal,
  modelMethods = "SVC", outputModel = FALSE, nIter = 7000, burnIn = 500,
  thin = 5, saveAt = "", S0 = NULL, df0 = 5, R2 = 0.5,
  weights = NULL, verbose = FALSE, rmExistingFiles = TRUE,
  groups = NULL, importance = FALSE, posPercentage = 0.4,
  BestIndividuals = c("top"), ntree = 500, nodesize = 1,
  kernel = c("linear"), gamma = 1, cost = 2^(-9), K = 8, eta = 0.7,
  select = "pls2", fit = "simpls", scale.x = FALSE, scale.y = FALSE,
  eps = 1e-04, trace = FALSE, maxstep = 100, alpha = 1, X = NULL,
  family = gaussian(link = identity), lambda = NULL, tol.err = 1e-06,
  tol.conv = 1e-08, epochs = 20, neurons = 4, Z = NULL,
  effectConcider = "A", mmerIters = 8, ...)
```

```
G2PCrossValidation(cross = 5, seed = 1, cpus = 1, markers, pheVal,
  modelMethods = "SVC", outputModel = FALSE, nIter = 7000, burnIn = 500,
  thin = 5, saveAt = "", S0 = NULL, df0 = 5, R2 = 0.5,
  weights = NULL, verbose = FALSE, rmExistingFiles = TRUE,
  groups = NULL, importance = FALSE, posPercentage = 0.4,
  BestIndividuals = c("top"), ntree = 500, nodesize = 1,
  kernel = c("linear"), gamma = 1, cost = 2^(-9), K = 8, eta = 0.7,
  select = "pls2", fit = "simpls", scale.x = FALSE, scale.y = FALSE,
  eps = 1e-04, trace = FALSE, maxstep = 100, alpha = 1, X = NULL,
  family = gaussian(link = identity), lambda = NULL, tol.err = 1e-06,
  tol.conv = 1e-08, epochs = 20, neurons = 4, Z = NULL,
  effectConcider = "A", mmerIters = 8, ...)
```

Arguments

`cross` (numeric)the fold number of cross validation.

`seed` (numeric)random number options,defult 1.

`cpus` (numeric)number of cpu cores to be used for calculations.

`markers` (numeric) a matrix, each row is the each individual's SNP genotypes informations.Genotypes should be coded as 0,1,2;0 represents AA(homozygote),2 represents BB(homozygote) and 1 represents AB(heterozygote);missing (NA) alleles are not allowed `pheVal` (numeric)the phenotype Value of each individual.

`modelMethods` the model to fit."BayesA", "BayesB", "BayesC", "BL", "BRR","RKHS","rrBLUP","LASSO","SPLS","SV
"RR", "RKHS", "BRNN", "EM", "EMMA", "AI", "NR".

`outputModel` if true return the list of training model.

`nIter, burnIn, thin`
(integer) the number of iterations, burn-in and thinning,default nIter 7000,burnIn 500,thin 5.

`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,default ""

`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.Default S0 NULL,df0 5.

`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-extdoc.pdf.Defult 0.5

`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.

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

`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.

`importance` RandomForest parameter:Should importance of predictors be assessed?Defualt FALSE

`posPercentage` (numeric)the percentage positive samples in all samples.1 > posPercentage > 0.

`BestIndividuals`
It is a position that the best individuals (positive samples) in a training group, according to the breeding values of a training group's trait. if the trait was yield,flowering or disease resistance,and male flowering time to female flowering time,it is "top"(default), "buttom",and "middle" of the breeding values, respectively.

`ntree` RandomForest parameter:Number of trees to grow. This should not be set to too small a number, to ensure that every input row gets predicted at least a few times.Defualt 500 `nodesize` Randomforest parameter Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). Note that the default values are different for classification (1) and regression (5).

`kernel` svm parameter the kernel used in training and predicting. You might consider changing some of the following parameters, depending on the kernel type.(linear,polynomial,sigmoid,radi "linear".

`gamma` svm parameter parameter needed for all kernels except linear (default: 1/(data dimension))

`cost` svm cost,default 2^(-9)

`K` Number of hidden components

`eta` Thresholding parameter. eta should be between 0 and 1.

`select` PLS algorithm for variable selection. Alternatives are "pls2" or "simpls". Default is "pls2"

`fit` PLS algorithm for model fitting. Alternatives are "kernelpls", "widekernelpls", "simpls", or "oscorespls". Default is "simpls".

`scale.x` Scale predictors by dividing each predictor variable by its sample standard deviation?

`scale.y` Scale responses by dividing each response variable by its sample standard deviation?

`eps` An effective zero. Default is 1e-4

`trace` Print out the progress of variable selection?

`maxstep` Maximum number of iterations when fitting direction vectors. Default is 100.

`alpha` The elasticnet mixing parameter.Detail in glmnet.

`X` (numeric, matrix) design matrix related to the parameters not to be shrunk (i.e. fixed effects in the mixed model framework),defult no shrink.

`family` the distribution family of y, see help('family') for more details.

`lambda` the shrinkage parameter determines the amount of shrinkage. Default is NULL meaning that it is to be estimated along with other model parameters.

`tol.err` internal tolerance level for extremely small values; default value is 1e-6.

`tol.conv` tolerance level in convergence; default value is 1e-8.

`epochs` (integer)maximum number of epochs(iterations) to train, default 30.

`neurons` (integer)indicates the number of neurons,defult 4.

`Z` (2-level list)incidence matrices and var-cov matrices for random effects. This works for ONE OR MORE random effects. This needs to be provided as a 2-level list structure, defult NULL.

`effectConcider` (string)if Z = NULL, random effects are auto generated.

`...` arguments passed to or from other methods.

`iters` (numeric)a scalar value indicating how many iterations have to be performed if the optimization algorithms. There is no rule of tumb for the number of iterations but less than 8 is usually enough, default 20.

`cross` (numeric)the fold number of cross validation.

`seed` (numeric)random number options,defult 1.

`cpus` (numeric)number of cpu cores to be used for calculations.

`markers` (numeric) a matrix, each row is the each individual's SNP genotypes informations.Genotypes should be coded as 0,1,2;0 represents AA(homozygote),2 represents BB(homozygote) and 1 represents AB(heterozygote);missing (NA) alleles are not allowed

`pheVal` (numeric)the phenotype Value of each individual.

`modelMethods` the model to fit."BayesA", "BayesB", "BayesC", "BL", "BRR","RKHS","rrBLUP","LASSO","SPLS","SV "RR", "RKHS", "BRNN", "EM", "EMMA", "AI", "NR".

`nIter, burnIn, thin`
(integer) the number of iterations, burn-in and thinning,default nIter 7000,burnIn 500,thin 5.

`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,default ""

`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.Default S0 NULL,df0 5.

`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-extdoc.pdf.Defult 0.5

`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.

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

`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.

`ntree` RandomForest parameter:Number of trees to grow. This should not be set to too small a number, to ensure that every input row gets predicted at least a few times.Defualt 500

`nodesize` Randomforest parameter Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). Note that the default values are different for classification (1) and regression (5).

`importance` RandomForest parameter:Should importance of predictors be assessed?Defualt FALSE `posPercentage` (numeric)the percentage positive samples in all samples.1 > posPercentage > 0.

`BestIndividuals`
It is a position that the best individuals (positive samples) in a training group, according to the breeding values of a training group's trait. if the trait was yield,flowering or disease resistance,and male flowering time to female flowering time,it is "top"(default), "buttom",and "middle" of the breeding values, respectively.

`kernel` svm parameter the kernel used in training and predicting. You might consider changing some of the following parameters, depending on the kernel type.(linear,polynomial,sigmoid,radi
"linear".

`gamma` svm parameter parameter needed for all kernels except linear (default: 1/(data dimension))

`cost` svm cost,default 2^(-9)

`outputModel` if true return the list of training model.

`K` Number of hidden components

`eta` Thresholding parameter. eta should be between 0 and 1.

`select` PLS algorithm for variable selection. Alternatives are "pls2" or "simpls". Default is "pls2"

`fit` PLS algorithm for model fitting. Alternatives are "kernelpls", "widekernelpls", "simpls", or "oscorespls". Default is "simpls".

`scale.x` Scale predictors by dividing each predictor variable by its sample standard deviation?

`scale.y` Scale responses by dividing each response variable by its sample standard deviation?

`eps` An effective zero. Default is 1e-4

`maxstep` Maximum number of iterations when fitting direction vectors. Default is 100.

`trace` Print out the progress of variable selection?

`alpha` The elasticnet mixing parameter.Detail in glmnet.

`X` (numeric, matrix) design matrix related to the parameters not to be shrunk (i.e. fixed effects in the mixed model framework),defult no shrink.

`family` the distribution family of y, see help('family') for more details.

`lambda` the shrinkage parameter determines the amount of shrinkage. Default is NULL meaning that it is to be estimated along with other model parameters.

`tol.err` internal tolerance level for extremely small values; default value is 1e-6.

`tol.conv` tolerance level in convergence; default value is 1e-8.

`neurons` (integer)indicates the number of neurons,defult 4.

`epochs` (integer)maximum number of epochs(iterations) to train, default 30.

`Z` (2-level list)incidence matrices and var-cov matrices for random effects. This works for ONE OR MORE random effects. This needs to be provided as a 2-level list structure, defult NULL.

`effectConcider` (string)if Z = NULL, random effects are auto generated.

`iters` (numeric)a scalar value indicating how many iterations have to be performed if the optimization algorithms. There is no rule of tumb for the number of iterations but less than 8 is usually enough, default 20.

`...` arguments passed to or from other methods.

Value

a list: The prediction results of input GS method with cross validation.

a list: The prediction results of input GS method with cross validation.

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

See Also

GSmachine GSReModel fit.RR fit.BRNN fit.RKHS fit.mmer predictGS G2P

GSmachine GSReModel fit.RR fit.BRNN fit.RKHS fit.mmer predictGS G2P

Examples

```
## Not run:
data(GYSS)
########## predicting breeding value
predlist <- G2PCrossValidation(cross = 10,seed = 1 , cpus = 3, markers = Markers,
                pheVal = phenotype, modelMethods = c("rrBLUP", "RFC"),
                outputModel = FALSE)

## End(Not run)
```

```
## Not run:
data(GYSS)
########## predicting breeding value
predlist <- G2PCrossValidation(cross = 10,seed = 1 , cpus = 3, markers = Markers,
                pheVal = phenotype, modelMethods = c("rrBLUP", "RFC"),
                outputModel = FALSE)

## End(Not run)
```

`G2PTest` *G2PTest*

Description

Testing the program and get the time of one cross validation.

Usage

```
G2PTest(cross = 10, ncross = NULL, seed = 1, cpus = 1, markers, pheVal,
  modelMethods = "SVC", nIter = 7000, burnIn = 500, thin = 5,
  saveAt = "", S0 = NULL, df0 = 5, R2 = 0.5, weights = NULL,
  verbose = FALSE, rmExistingFiles = TRUE, groups = NULL,
  importance = FALSE, posPercentage = 0.4, BestIndividuals = c("top"),
  ntree = 500, nodesize = 1, kernel = c("radial"), gamma = 1,
  cost = 2^(-9), outputModel = FALSE, K = 8, eta = 0.7,
  select = "pls2", fit = "simples", scale.x = FALSE, scale.y = FALSE,
  eps = 1e-04, trace = FALSE, maxstep = 100, alpha = 1, X = NULL,
  family = gaussian(link = identity), lambda = NULL, tol.err = 1e-06,
  tol.conv = 1e-08, epochs = 20, neurons = 4, Z = NULL,
  effectConcider = "A", mmerIters = 8, ...)
```

Arguments

`cross` (numeric)the fold number of cross validation.

`seed` (numeric)random number options,defult 1.

`cpus` (numeric)number of cpu cores to be used for calculations.

`markers` (numeric) a matrix, each row is the each individual's SNP genotypes informations.Genotypes should be coded as 0,1,2;0 represents AA(homozygote),2 represents BB(homozygote) and 1 represents AB(heterozygote);missing (NA) alleles are not allowed

`pheVal` (numeric)the phenotype Value of each individual.

`modelMethods` the model to fit."BayesA", "BayesB", "BayesC", "BL", "BRR","RKHS","rrBLUP","LASSO","SPLS","SV "RR", "RKHS", "BRNN", "EM", "EMMA", "AI", "NR".

`nIter, burnIn, thin`
(integer) the number of iterations, burn-in and thinning,default nIter 7000,burnIn 500,thin 5.

`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,default ""

`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.Default S0 NULL,df0 5.

`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-extdoc.pdf.Defult 0.5

`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.

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

`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.

`importance` RandomForest parameter:Should importance of predictors be assessed?Defualt FALSE

`posPercentage` (numeric)the percentage positive samples in all samples.1 > posPercentage > 0.

`BestIndividuals` It is a position that the best individuals (positive samples) in a training group, according to the breeding values of a training group's trait. if the trait was yield,flowering or disease resistance,and male flowering time to female flowering time,it is "top"(default), "buttom",and "middle" of the breeding values, respectively.

`ntree` RandomForest parameter:Number of trees to grow. This should not be set to too small a number, to ensure that every input row gets predicted at least a few times.Defualt 500

`nodesize` Randomforest parameter Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). Note that the default values are different for classification (1) and regression (5).

`kernel` svm parameter the kernel used in training and predicting. You might consider changing some of the following parameters, depending on the kernel type.(linear,polynomial,sigmoid,radi "linear".

`gamma` svm parameter parameter needed for all kernels except linear (default: 1/(data dimension))

`cost` svm cost,default 2^(-9)

`outputModel` if true return the list of training model.

`K` Number of hidden components

`eta` Thresholding parameter. eta should be between 0 and 1.

`select` PLS algorithm for variable selection. Alternatives are "pls2" or "simpls". Default is "pls2"

`fit` PLS algorithm for model fitting. Alternatives are "kernelpls", "widekernelpls", "simpls", or "oscorespls". Default is "simpls".

`scale.x` Scale predictors by dividing each predictor variable by its sample standard deviation?

`scale.y` Scale responses by dividing each response variable by its sample standard deviation?

`eps` An effective zero. Default is 1e-4

`trace` Print out the progress of variable selection?

`maxstep` Maximum number of iterations when fitting direction vectors. Default is 100.

`alpha` The elasticnet mixing parameter.Detail in glmnet.

`X` (numeric, matrix) design matrix related to the parameters not to be shrunk (i.e. fixed effects in the mixed model framework),defult no shrink.

`family` the distribution family of y, see help('family') for more details.

`lambda` the shrinkage parameter determines the amount of shrinkage. Default is NULL meaning that it is to be estimated along with other model parameters.

`tol.err` internal tolerance level for extremely small values; default value is 1e-6.

`tol.conv` tolerance level in convergence; default value is 1e-8.

`epochs` (integer)maximum number of epochs(iterations) to train, default 30.

`neurons` (integer)indicates the number of neurons,defult 4.

`Z` (2-level list)incidence matrices and var-cov matrices for random effects. This works for ONE OR MORE random effects. This needs to be provided as a 2-level list structure, defult NULL.

`effectConcider` (string)if Z = NULL, random effects are auto generated.

`...` arguments passed to or from other methods.

`iters` (numeric)a scalar value indicating how many iterations have to be performed if the optimization algorithms. There is no rule of tumb for the number of iterations but less than 8 is usually enough, default 20.

Value

a list: $runOneFoldTime The prediction results of input GS method with cross validation. $res Test program results

Author(s)

Chuang Ma ,Qian Cheng , Zhixu Qiu ,Jie Song

See Also

GSmachine GSReModel fit.RR fit.BRNN fit.RKHS fit.mmer predictGS G2P

Examples

```
## Not run:
data(GYSS)
########## predicting breeding value
test <-  G2PTest(cross = 10, seed = 1, cpus = 3, markers = Markers,
                 pheVal = phenotype, modelMethods = c("rrBLUP", "RFC"),
                 outputModel = FALSE)

## End(Not run)
```

GSDataQC *Genomic Selection Data Quality control*

Description

This function can examine and summary the quality of GS data. And can be used for imputation.

Usage

```
GSDataQC(markers, phenotype, impute = F, imputeMethod = "mean", round = 4,
  k = 10, maxp = 1500, rowmax = 0.5, colmax = 0.8,
  rng.seed = 362436069)
```

Arguments

`markers`	(numeric, matrix)row is sample well column is SNP information (feature).Genotypes should be coded as 0,1,2;0 represent AA(homozygote),2 represent BB(homozygote) and 1 represent AB(heterozygote);missing (NA) alleles are not allowed.
`phenotype`	(numeric)the phenotype value of each individual.
`impute`	(logical)if TRUE, imputation, default F.
`imputeMethod`	(character)the method of imputation, "mean", "median" or "KNN", default "mean".
`round`	(numeric)rounds the values in its first argument to the specified number of decimal places, default 4.
`k, maxp, rowmax, colmax, rng.seed`	(numeric)KNN method parameters.

Value

A list of the data quality information.

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

See Also

impute

Examples

```
## Not run:
data(GYSS)
## generate missing value
misIndex <- sample(1:242000,100000)
Markers[misIndex] <- NA

## GSDataQC, not impute ##
```

```
QCRes <- GSDataQC(markers = Markers, phenotype = phenotype, impute = F)

## GSDataQC, not impute ##
QCResImpute <- GSDataQC(markers = Markers, phenotype = phenotype, impute = T,
                                    imputeMethod = "mean")

## End(Not run)
```

GSEnsemble *Methods Ensemble From Two or More Methods*

Description

This function provides a strategy to ensemble the results of two or more algorithms. It is a extension of GSIntegrate

Usage

```
GSEnsemble(predMat, nrandom = 10, evalMethods, by = 0.1, evaluation = T,
  topAlpha = 15, ...)
```

Arguments

`predMat`	(numeric, matrix)the prediction results of algorithms which you want to merge, the first column is the real value of trait.
`nrandom`	(integer)the repeat number of stacking, default 10.
`evalMethods`	(character)ensemble base which evaluation methods.
`by`	(numeric,(0,1))the radio window of ensemble, the smaller "by", the higher accuracy of ensemble. Default 0.1.
`evaluation`	(logical)if evaluate finalMat with evalMethods, default TRUE.
`topAlpha`	(numeric)the parameter of threshold evaluation methods, see also function evaluateGS. In this function, indicates the best ensemble base threshold when evalMethods is threshold methods.
`...`	arguments passed to or from other functions.

Value

a list: $BestWeight The best weight of methods in all repeat $finalMat The final matrix cbind predMat with final ensemble score $evalRes The evaluation results of finalMat with evalMethods $weightMat A weight matrix including all repeats $evalMat A evaluation results matrix including all repeats

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

See Also

GSIntegrate evaluateGS

Examples

```
## Not run:
## Load example data ##
data(GYSS)

## cross validation ##
predlist <- G2PCrossvalidation(cross = 10,seed = 1 , cpus = 3, markers = Markers,
             pheVal = phenotype, modelMethods = c("BayesA","BayesB","BayesC","rrBLUP", "RFR"),
                outputModel = FALSE)
resultMat <- resultsMerge(predlist)

## merge ##
ensembleRes <- GSEnsemble(predMat = resultMat, nrandom = 10, evalMethods = "RE",
                          by = 0.1, evaluation = T, topAlpha = 15 )

## End(Not run)
```

GSIntegrate *Prediction Integration from Two Methods*

Description

This function provides a strategy to integrate the results of two or more algorithms.

Usage

```
GSIntegrate(predResMat, ratio, autoOptimize = F)
```

Arguments

predResMat	(numeric, matrix)the prediction results of algorithms which you want to merge, the first column is the real value of trait.
ratio	(numeric,array) the weights of every algorithms.
autoOptimize	(logical)if auto select two method results from multi-results and then compute the mean of two methods results (1:1), default FALSE.

Details

The predResMat must including real value in first clumn, and if you set "autoOptimize = T", the count of algorithms must more than 2.

In this function, if autoOptimize = T, the final two algorithms merge are selected from multi methods by following strategy: Firstly, compute the pearson's correlation of predResMat, choose the best correlation between real value and prediction scores, named method 1. Secondly, choose the best correlation between method 1 and other methods, named method 2. Finally, merge method 1 and method 2 with 1:1(mean).

This function auto merge only provide integrate base **pearson's correlation** evaluation.

Value

a matrix: involve real value of trait, merge algorithms and the merge result.

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

See Also

GSEnsemble

Examples

```
## Not run:
## Load example data ##
data(GYSS)

## cross validation ##
predlist <- G2PCrossvalidation(cross = 10,seed = 1 , cpus = 3, markers = Markers,
                pheVal = phenotype, modelMethods = c("BayesA","BayesB","BayesC","rrBLUP", "RFR"),
                   outputModel = FALSE)
resultMat <- resultsMerge(predlist)

## merge ##
inter <- GSIntegrate(predResMat = resultMat[,1:6],
                     ratio = c(2,3,4,4,5), autoOptimize = F)
interAuto <- GSIntegrate(predResMat = resultMat, autoOptimize = T,
                         allMethodPredResMat = resultMat)

## End(Not run)
```

GSmachine *Fit machine learning model*

Description

This function can fit several machine learning models of genomic selection such as "SVR", "SVC", "RFR" and "RFC".

Usage

```
GSmachine(markers, pheVal, modelMethods = "SVC", posPercentage = 0.4,
  BestIndividuals = c("top"), ntree = 500, nodesize = 1,
  kernel = c("linear"), gamma = 1, cost = 2^(-9), ...)
```

Arguments

`markers` (numeric)a matrix, each row is the each individual's SNP genotypes informations.Genotypes should be coded as 0,1,2;0 represent AA(homozygote),2 represent BB(homozygote) and 1 represent AB(heterozygote);missing (NA) alleles are not allowed.

`pheVal` (numeric)the phenotype value of each individual.

`modelMethods` (character)alternative machine learning models. "SVR" and "SVC" from SVM, "RFR" and "RFC" from RF.

`posPercentage` (numeric,[0,1])phenotype of extreme individuals which expected, default 0.4.

`BestIndividuals`
(character)the position of expected phenotype in whole phenotypic data set."top","buttom" or "middle",default "top".

`ntree` (integer)ramdomforest parameter. Number of trees to grow. This should not be set to too small a number, to ensure that every input row gets predicted at least a few times,default 500.

`nodesize` (integer)ramdomforest parameter Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). Note that the default values are different for classification (1) and regression (5).

`kernel` (numeric)svm parameter the kernel used in training and predicting. You might consider changing some of the following parameters, depending on the kernel type.(linear,polynomial,sigmoid,radial)Default "linear".

`gamma` (numeric)svm parameter parameter needed for all kernels except linear, default 1.

`cost` (numeric)svm parameter cost of constraints violation, default: 2^(-9), it is the 'C'-constant of the regularization term in the Lagrange formulation.

`...` other parameters.

Details

SVM (support vector machine) and RF (random forest) models can be fitted in this function, including 2 classification (RFC, SVC) and 2 regression (SVR, SVC) models.

Value

a machine learning model

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

See Also

e1017 randomForest

Examples

```
## Not run:
## Load example data ##
data(GYSS)

## Fit RFR model ##
machine_model <- GSmachine(markers = Markers, pheVal = phenotype, modelMethods = "RFR")

## Fit classification model(RFC) ##
machine_model <- GSmachine(markers = Markers, pheVal = phenotype, modelMethods = "RFC",
                           posPercentage = 0.4, ntree = 500)

## End(Not run)
```

GSReModel *Fit Regression Model*

Description

This function can fit several regression models of genomic selection including SPLS, rrBLUP, LASSO and bigRR.

Usage

```
GSReModel(markers, pheVal, modelMethods, K = 8, eta = 0.7,
  select = "pls2", fit = "simpls", scale.x = FALSE, scale.y = FALSE,
  eps = 1e-04, trace = FALSE, maxstep = 100, alpha = 1, X = NULL,
  family = gaussian(link = identity), lambda = NULL, tol.err = 1e-06,
  tol.conv = 1e-08, weights = NULL, ...)
```

Arguments

`markers`	(numeric)a matrix, each row is the each individual's SNP genotypes informations.Genotypes should be coded as 0,1,2or-1,0,1;0(-1) represent AA(homozygote),2(1) represent BB(homozygote) and 1(0) represent AB(heterozygote); missing (NA) alleles are not allowed.
`pheVal`	(numeric)the phenotype value of each individual.
`K`	(integer)SPLS model parameter: number of hidden components.
`eta`	(numeric)SPLS model parameter: thresholding parameter. eta should be between 0 and 1.
`select`	(character)SPLS model parameter: PLS algorithm for variable selection. Alternatives are "pls2" or "simpls". Default is "pls2".
`fit`	(character)SPLS model parameter: PLS algorithm for model fitting. Alternatives are "kernelpls", "widekernelpls", "simpls", or "oscorespls". Default is "simpls".

`scale.x` (character)SPLS model parameter: scale predictors by dividing each predictor variable by its sample standard deviation?

`scale.y` (character)SPLS model parameter: scale responses by dividing each response variable by its sample standard deviation?

`eps` (character)SPLS model parameter: an effective zero. Default is 1e-4.

`trace` (logical)SPLS model parameter: print out the progress of variable selection?

`maxstep` (integer)SPLS model parameter: maximum number of iterations when fitting direction vectors. Default is 100.

`alpha` (numeric)LASSO model parameter: the elasticnet mixing parameter.Detail in glmnet.

`X` (numeric, matrix) design matrix related to the parameters not to be shrunk (i.e. fixed effects in the mixed model framework),defult no shrink.

`family` the distribution family of y, see help('family') for more details.

`lambda` the shrinkage parameter determines the amount of shrinkage. Default is NULL meaning that it is to be estimated along with other model parameters.

`tol.err` internal tolerance level for extremely small values; default value is 1e-6.

`tol.conv` tolerance level in convergence; default value is 1e-8.

`...` arguments passed to or from other package.

Value

A regression model which is enable to predict.

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

See Also

rrBLUP hglm glmnet spls

Examples

```
## Not run:
## Load example data ##
data(GYSS)

## Fit rrBLUP model ##
rrBLUP_model <- GSReModel(markers = Markers,pheVal = phenotype,modelMethods = "rrBLUP")

## End(Not run)
```

GYSS *Example Data for G2P*

Description

The data of maize yield under drought stressed.(SNP genotypes informations)

- Markers A numeric matrix, each row is the each individual’s SNP genotypes informations.
- phenotype The real phenotype Value of each individual.

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

Examples

```
## Not run:
## load maize yield data sets
data(GYSS)

## End(Not run)
```

Markers *Example Data for G2P*

Description

A numeric matrix.

- each row represents each sample
- each column represents each SNP locus

Format

matrix

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

phenotype *Example Data for G2P*

Description

The real phenotype Value of each individual

Format

vector

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

predictGS *Prediction with Trained Model from Geomic Selection Model*

Description

This function can give the prediction score of a new GS data by using already model.

Usage

```
predictGS(testMat, trainModel, modelMethods = "SVC", type = "fit")
```

Arguments

`testMat` (numeric)a matrix, each row is the each testing sets or new GS data individual's SNP genotypes informations.Genotypes should be coded as 0,1,2; 0 represent AA(homozygote), 2 represent BB(homozygote) and 1 represent AB(heterozygote); missing (NA) alleles are not allowed.

`trainModel` (model)the trained model.It's type must be similar whith modelMethods.

`modelMethods` (character)the type name of training model including "bigRR","rrBLUP","LASSO","SPLS","SVC","SVR

`type` (character)SPLS parameter,detail see package spls.**spls**.

Value

a list: The prediction result of testing sets which predicted through already models

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

Examples

```
## Not run:
## Load example data ##
data(GYSS)

## Fit rrBLUP model ##
rrBLUP_model <- GSReModel(markers = Markers, pheVal = phenotype, modelMethods = ”rrBLUP”)

## Predict 1-20 subset of all example data with already rrBLUP model ##
res <- predictGS(testMat = Markers[1:20,], trainModel = rrBLUP_model, modelMethods = ”rrBLUP”)

## End(Not run)
```

`randomSeed` *Generate Random Seed*

Description

This funcation is appplied for generating random seed with current system time

Usage

```
randomSeed()
```

Value

(numeric) A random seed

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

`resultsMerge` *Transform Prediction Result List to Matrix*

Description

This function provides a way to get prediction matix from prediction list.

Usage

```
resultsMerge(predList)
```

Arguments

`predList`	prediction list.

Value

the prediction result matrix.

Author(s)

Chuang Ma , Qian Cheng , Zhixu Qiu , Jie Song

Examples

```
## Not run:
## Load example data ##
data(GYSS)

## cross validation ##
predlist <- G2PCrossvalidation(cross = 10,seed = 1 , cpus = 3, markers = Markers,
             pheVal = phenotype, modelMethods = c("BayesA","BayesB","BayesC","rrBLUP", "RFC"),
                outputModel = FALSE)
resultMat <- resultsMerge(predlist)

## End(Not run)
```

`rowDataPlot` *The Visualization of Evaluation Result and Data Structure*

Description

This function is designed for visualization of multiple results of G2P.

Usage

```
rowDataPlot(y,show.line = T,barCol = "blue",lineCol = "red")
       rowDataPlot(markers,y,plot.type = "PCA")
      scatterPlot(predmat,x1 ,x2 = ,show.line = F,color_szie = T,make.plotly = F,sizeRange = c(4,6))
       linePlot(evalMat,size = 1)
       barPlot(data,other = "sector")
      heatmapEval <- heatMapDataProcess(x,highBound = 0,lowBound = -30,alpha = 15, basedMethod = "best
```

Value

plotd of result

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

Examples

```
## Not run:
############# PCA analysis ############
data(GYSS)
G2PCVRes <-  G2PCrossValidation(cross = 10,seed = 1 , cpus = 3, markers = Markers,
pheVal = phenotype, modelMethods = c("BayesA", "BayesB", "BayesC", "BL", "BRR","RR",
                    "RKHS","rrBLUP","LASSO","SPLS","bigRR","SVC","RFC","SVR","RFR"),
outputModel = FALSE)
CVres <- resultsMerge(predList = G2PCVRes)
evalTest <- evaluateGS(realScores = CVres[,1], predScores = CVres[,2:20],
                      evalMethod = c( "pearson", "kendall","spearman", "RE", "Kappa",
                                      "AUC", "AUCpr", "NDCG", "meanNDCG",
                                      "MSE", "R2", "F1", "accuracy"), topAlpha = 1:90)
### row data visulization
## phenotype distribution  plot
rowDataPlot(y = phenotype,show.line = T,barCol = "blue",lineCol = "red")
## PCA 3-D plot
htmlwidgets::saveWidget(as_widget(rowDataPlot(markers = Markers,y = phenotype,
                      plot.type = "PCA")), file="3-D_PCA.html",selfcontained=T)

### scatter plot
scatterPlot(CVres,x1 = "BayesA",x2 = "RFC",show.line = F,color_szie = T,make.plotly = F,
           sizeRange = c(4,6))
### lines plot
linePlot(evalMat = evalTest$RE,size = 1)
### bar plot
barPlot(evalTest$corMethosds,other = "sector")
### heat map
#### pred res heatmap
heatmapPlot(predmat = CVres,make.plotly = F,col.low = "green",col.high = "red")
#### eval res heatmap
heatmapEval <- heatMapDataProcess(x = evalTest,highBound = 0,lowBound = -30,alpha = 15,
                                 basedMethod = "best")
heatmapPlot(predmat = heatmapEval,make.plotly = F,col.low = "green",col.high = "red")

## End(Not run)
```

---

sampleClassify *Generate Positive and Negative Samples for Training*

Description

This function can be use to generate positive and negative samples for training.The positive samples represent the excellent individuals which's breeding values we expect to obtain in your research.And the negative samples represent the lower breeding values of individuals.

Usage

```
sampleClassify(phenotype, posPercentage = 0.4, BestIndividuals = c("top",
  "middle", "buttom"))
```

Arguments

`phenotype` (numeric)the breeding values of each individual.

`posPercentage` (numeric,[0,1])phenotype of extreme individuals which expected, default 0.4

`BestIndividuals` (character)the position of expected phenotype in whole phenotypic data set."top","buttom" or "middle",default "top".

Value

A list of row number of positive and negative samples $posSampleIndex Index of positive samples $negSampleIndex Index of negative samples

Author(s)

Chuang Ma, Qian Cheng, Zhixu Qiu, Jie Song

Examples

```
## Not run:
data(GYSS)
## percentage of positive samples is 0.4 ##
sampleCly <- sampleClassify(phenotype, posPercentage = 0.4, BestIndividuals = "top")

## End(Not run)
```

Index