Package 'factReg'

July 6, 2023

Type Package

Title Multi-Environment Genomic Prediction with Penalized Factorial Regression

Version 1.0.0

Date 2023-07-05

Description Multi-environment genomic prediction for training and test environments using penalized factorial regression. Predictions are made using genotype-specific environmental sensitivities as in Millet et al. (2019) <doi:10.1038/s41588-019-0414-y>.

License GPL-3

Depends R (>= 3.6)

Imports glmnet,
mathjaxr,
Matrix,
rrBLUP,
stats

Encoding UTF-8

LazyData true

LazyDataCompression gzip

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE)

RdMacros mathjaxr

Suggests tinytest

R topics documented:

drops_GE *DROPS data sets*

Description

These datasets come from the European Union project DROPS (DROught-tolerant yielding PlantS). A panel of 256 maize hybrids was grown with two water regimes (irrigated or rainfed), in seven fields in 2012 and 2013, respectively, spread along a climatic transect from western to eastern Europe, plus one site in Chile in 2013. This resulted in 28 experiments defined as the combination of one year, one site and one water regime, with two and three repetitions for rainfed and irrigated treatments, respectively. A detailed environmental characterisation was carried out, with hourly records of micrometeorological data and soil water status, and associated with precise measurement of phenology. Grain yield and its components were measured at the end of the experiment.
The data sets contain the genotypic BLUEs for eight traits for 246 genotypes in 37 environments. Additionally information on 11 environmental indices is included. The environments are split in three data sets for training (drops_GE) and testing (drops_GnE, drops_nGnE) purposes. drops_K contains the kinship matrix for the 246 genotypes.

Usage

```
drops_GE

drops_GnE

drops_nGnE

drops_K
```

Format

data.frames with 24 variables.

Experiment experiments ID described by the three first letters of the city's name followed by the year of experiment and the water regime with W for watered and R for rain-fed.

Code_ID, Variety_ID, Accession_ID identifier of the genotype

grain.yield genotypic mean for yield adjusted at 15\ in ton per hectare (t ha^-1)

grain.number genotypic mean for number of grain per square meter

grain.weight genotypic mean for individual grain weight in milligram (mg)

anthesis genotypic mean for male flowering (pollen shed), in thermal time cumulated since emergence (d_20°C)

silking genotypic mean for female flowering (silking emergence), in thermal time cumulated since emergence (d_20°C)

plant.height genotypic mean for plant height, from ground level to the base of the flag leaf (highest) leaf in centimeter (cm)

tassel.height genotypic mean for plant height including tassel, from ground level to the highest point of the tassel in centimeter (cm)

ear.height genotypic mean for ear insertion height, from ground level to ligule of the highest ear leaf in centimeter (cm)

`Tnight.Early` night temperature averaged between the floral transition and the silk initiation

`Tnight.Flo` night temperature averaged between the silk initiation and the end of grain abortion

`Tnight.Fill` night temperature averaged between the end of grain abortion and the physiological maturity of the grain

`Ri.Early` intercepted radiation cumulated between the floral transition and the silk initiation

`Ri.Flo` intercepted radiation cumulated between the silk initiation and the end of grain abortion

`Ri.Fill` intercepted radiation cumulated between the end of grain abortion and the physiological maturity of the grain

`Psi.Flo` soil water potential averaged between the silk initiation and the end of grain abortion. The soil water potential used here was the median between 30 and 60cm depth.

`Psi.Fill` soil water potential averaged between the end of grain abortion and the physiological maturity of the grain

`Tmax.Early` maximum temperature averaged between the floral transition and the silk initiation

`Tmax.Flo` maximum temperature averaged between the silk initiation and the end of grain abortion

`Tmax.Fill` maximum temperature averaged between the end of grain abortion and the physiological maturity of the grain

`type` code corresponding to the data set. GE for drops_GE, GnE for drops_GnE and nGnE for drops_nGnE.

An object of class `data.frame` with 384 rows and 24 columns.

An object of class `data.frame` with 224 rows and 24 columns.

An object of class `matrix` (inherits from `array`) with 302 rows and 302 columns.

Source

doi:10.15454/IASSTN

References

Millet, E. J., Pommier, C., et al. (2019). A multi-site experiment in a network of European fields for assessing the maize yield response to environmental scenarios (Data set). doi:10.15454/IASSTN

`GnE` *Penalized factorial regression using glmnet*

Description

Based on multi-environment field trials, fits the factorial regression model $Y_{ij} = \mu + e_j + g_i + \sum_{k=1}^{s} \beta_{ik} x_{ij} + \epsilon_{ij}$, with environmental main effects e_j, genotypic main effects g_i and genotype-specific environmental sensitivities β_{ik}. See e.g. Millet et al 2019 and Bustos-Korts et al 2019. There are s environmental indices with values x_{ij}. Optionally, predictions can be made for a set of test environments, for which environmental indices are available. The new environments must contain the same set of genotypes, or a subset.

Penalization: the model above is fitted using glmnet, simultaneously penalizing e_j, g_i and β_{ik}. If `penG = 0` and `penE = 0`, the main effects g_i and e_j are not penalized. If these parameters are 1, the the main effects are penalized to the same degree as the sensitivities. Any non negative values are

allowed. Cross validation is performed with each fold containing a number of environments (details below).

After fitting the model, it is possible to replace the estimated genotypic main effects and sensitivities by their predicted genetic values. Specifically, if a kinship matrix K is assigned the function performs genomic prediction with g-BLUP for the genotypic main effect and each of the sensitivities in turn.

Predictions for the test environments are first constructed using the estimated genotypic main effects and sensitivities; next, predicted environmental main effects are added. The latter are obtained by regressing the estimated environmental main effects for the training environments on the average values of the indices in these environments, as in Millet et al. 2019.

Usage

```
GnE(
  dat,
  Y,
  G,
  E,
  K = NULL,
  indices = NULL,
  indicesData = NULL,
  testEnv = NULL,
  weight = NULL,
  outputFile = NULL,
  corType = c("pearson", "spearman"),
  partition = data.frame(),
  nfolds = 10,
  alpha = 1,
  lambda = NULL,
  penG = 0,
  penE = 0,
  scaling = c("train", "all", "no"),
  quadratic = FALSE,
  verbose = FALSE
)
```

Arguments

dat	A data.frame with data from multi-environment trials. Each row corresponds to a particular genotype in a particular environment. The data do not need to be balanced, i.e. an environment does not need to contain all genotypes. dat should contain the training as well as the test environments (see testEnv)
Y	The trait to be analyzed: either of type character, in which case it should be one of the column names in dat, or numeric, in which case the Yth column of dat will be analyzed.
G	The column in dat containing the factor genotype (either character or numeric).
E	The column in dat containing the factor environment (either character or numeric).
K	A kinship matrix. Used for replacing the estimated genotypic main effect and each of the sensitivities by genomic prediction from a g-BLUP model for each of them. If NULL, the estimated effects from the model are returned and used for constructing predictions.

`indices` The columns in `dat` containing the environmental indices (vector of type character). Alternatively, if the indices are always constant within environments (i.e. not genotype dependent), the environmental data can also be provided using the argument `indicesData` (see below).

`indicesData` An optional `data.frame` containing environmental indices (covariates); one value for each environment and index. It should have the environment names as row names (corresponding to the names contained in `dat$E`); the column names are the indices. If `indices` (see before) is also provided, the latter will be ignored.

`testEnv` vector (character). Data from these environments are not used for fitting the model. Accuracy is evaluated for training and test environments separately. The default is `NULL`, i.e. no test environments, in which case the whole data set is training. It is also possible that there are test environments, but without any data; in this case, no accuracy is reported for test environments (CHECK correctness).

`weight` Numeric vector of length `nrow(dat)`, specifying the weight (inverse variance) of each observation, used in glmnet. Default `NULL`, giving constant weights.

`outputFile` The file name of the output files, without .csv extension which is added by the function. If not `NULL` the prediction accuracies for training and test environments are written to separate files. If `NULL` the output is not written to file.

`corType` type of correlation: Pearson (default) or Spearman rank sum.

`partition` `data.frame` with columns E and partition. The column E should contain the training environments (type character); partition should be of type integer. Environments in the same fold should have the same integer value. Default is `data.frame()`, in which case the function uses a leave-one-environment out cross-validation. If `NULL`, the (inner) training sets used for cross-validation will be drawn randomly from all observations, ignoring the environment structure. In the latter case, the number of folds (nfolds) can be specified.

`nfolds` Default `NULL`. If `partition == NULL`, this can be used to specify the number of folds to be used in glmnet.

`alpha` Type of penalty, as in glmnet (1 = LASSO, 0 = ridge; in between = elastic net). Default is 1.

`lambda` Numeric vector; defines the grid over which the penalty lambda is optimized in cross validation. Default: NULL (defined by glmnet). Important special case: lambda = 0 (no penalty).

`penG` numeric; default 0. If 1, genotypic main effects are penalized. If 0, they are not. Any non negative real number is allowed.

`penE` numeric; default 0. If 1, environmental main effects are penalized. If 0, they are not. Any non negative real number is allowed.

`scaling` determines how the environmental variables are scaled. "train" : all data (test and training environments) are scaled using the mean and and standard deviation in the training environments. "all" : using the mean and standard deviation of all environments. "no" : No scaling.

`quadratic` boolean; default `FALSE`. If `TRUE`, quadratic terms (i.e., squared indices) are added to the model.

`verbose` boolean; default `FALSE`. If `TRUE`, the accuracies per environment are printed on screen.

Value

A list with the following elements:

predTrain A data.frame with predictions for the training set

predTest A data.frame with predictions for the test set

resTrain A data.frame with residuals for the training set

resTest A data.frame with residuals for the test set

mu the estimated overall mean

envInfoTrain The estimated environmental main effects, and the predicted effects, obtained when the former are regressed on the averaged indices, using penalized regression

envInfoTest The predicted environmental main effects for the test environments, obtained from penalized regression using the estimated main effects for the training environments and the averaged indices

parGeno data.frame containing the estimated genotypic main effects (first column) and sensitivities (subsequent columns)

trainAccuracyEnv a data.frame with the accuracy (r) for each training environment, as well as the root mean square error (RMSE), mean absolute deviation (MAD) and rank (the latter is a proportion: how many of the best 5 genotypes are in the top 10). To be removed or further developed. All these quantities are also evaluated for a model with only genotypic and environmental main effects (columns rMain, RMSEMain and rankMain)

testAccuracyEnv A data.frame with the accuracy for each test environment, with the same columns as trainAccuracyEnv

trainAccuracyGeno a data.frame with the accuracy (r) for each genotype, averaged over the training environments

testAccuracyGeno a data.frame with the accuracy (r) for each genotype, averaged over the test environments

lambda The value of lambda selected using cross validation

lambdaSequence The values of lambda used in the fits of glmnet. If `lambda` was provided as input, the value of `lambda` is returned

RMSEtrain The root mean squared error on the training environments

RMSEtest The root mean squared error on the test environments

Y The name of the trait that was predicted, i.e. the column name in `dat` that was used

G The genotype label that was used, i.e. the argument G that was used

E The environment label that was used, i.e. the argument E that was used

indices The indices that were used, i.e. the argument indices that was used

quadratic The quadratic option that was used

References

Millet, E.J., Kruijer, W., Coupel-Ledru, A. et al. Genomic prediction of maize yield across European environmental conditions. Nat Genet 51, 952–956 (2019). doi:10.1038/s415880190414y

Examples

```
## load the data, which are contained in the package
data(drops_GE)
data(drops_GnE)

## We remove identifiers that we don't need.
drops_GE_GnE <- rbind(drops_GE[, -c(2, 4)], drops_GnE[, -c(2, 4)])

## Define indeces.
ind <- colnames(drops_GE)[13:23]

## Define test environments.
testenv <- levels(drops_GnE$Experiment)

## Additive model, only main effects (set the penalty parameter to a large value).
Additive_model <- GnE(drops_GE_GnE, Y = "grain.yield", lambda = 100000,
                      G = "Variety_ID", E = "Experiment", testEnv = testenv,
                      indices = ind, penG = FALSE, penE = FALSE,
                      alpha = 0.5, scaling = "train")

## Full model, no penalization (set the penalty parameter to zero).
Full_model <- GnE(drops_GE_GnE, Y = "grain.yield", lambda = 0,
                  G = "Variety_ID", E = "Experiment", testEnv = testenv,
                  indices = ind, penG = FALSE, penE = FALSE,
                  alpha = 0.5, scaling = "train")

## Elastic Net model, set alpha parameter to 0.5.
Elnet_model <- GnE(drops_GE_GnE, Y = "grain.yield", lambda = NULL,
                   G = "Variety_ID", E = "Experiment", testEnv = testenv,
                   indices = ind, penG = FALSE, penE = FALSE,
                   alpha = 0.5, scaling = "train")

## Lasso model, set alpha parameter to 1.
Lasso_model <- GnE(drops_GE_GnE, Y = "grain.yield", lambda = NULL,
                   G = "Variety_ID", E = "Experiment", testEnv = testenv,
                   indices = ind, penG = FALSE, penE = FALSE,
                   alpha = 1, scaling = "train")

## Ridge model, set alpha parameter to 0.
Ridge_model <- GnE(drops_GE_GnE, Y = "grain.yield", lambda = NULL,
                   G = "Variety_ID", E = "Experiment", testEnv = testenv,
                   indices = ind, penG = FALSE, penE = FALSE,
                   alpha = 0, scaling = "train")
```

---

perGeno *Genomic prediction using glmnet, with a genotype-specific penalized regression model.*

Description

.... These models can be fitted either for the original data, or on the residuals of a model with only main effects.

Usage

```
perGeno(
  dat,
  Y,
  G,
  E,
  indices = NULL,
  indicesData = NULL,
  testEnv = NULL,
  weight = NULL,
  useRes = TRUE,
  outputFile = NULL,
  corType = c("pearson", "spearman"),
  partition = data.frame(),
  nfolds = 10,
  alpha = 1,
  scaling = c("train", "all", "no"),
  quadratic = FALSE,
  verbose = FALSE
)
```

Arguments

dat	A data.frame with data from multi-environment trials. Each row corresponds to a particular genotype in a particular environment. The data do not need to be balanced, i.e. an environment does not need to contain all genotypes. dat should contain the training as well as the test environments (see testEnv)
Y	The trait to be analyzed: either of type character, in which case it should be one of the column names in dat, or numeric, in which case the Yth column of dat will be analyzed.
G	The column in dat containing the factor genotype (either character or numeric).
E	The column in dat containing the factor environment (either character or numeric).
indices	The columns in dat containing the environmental indices (vector of type character). Alternatively, if the indices are always constant within environments (i.e. not genotype dependent), the environmental data can also be provided using the argument indicesData (see below).
indicesData	An optional data.frame containing environmental indices (covariates); one value for each environment and index. It should have the environment names as row names (corresponding to the names contained in dat$E); the column names are the indices. If indices (see before) is also provided, the latter will be ignored.
testEnv	vector (character). Data from these environments are not used for fitting the model. Accuracy is evaluated for training and test environments separately. The default is NULL, i.e. no test environments, in which case the whole data set is training. It is also possible that there are test environments, but without any data; in this case, no accuracy is reported for test environments (CHECK correctness).
weight	Numeric vector of length nrow(dat), specifying the weight (inverse variance) of each observation, used in glmnet. Default NULL, giving constant weights.

useRes	Indicates whether the genotype-specific regressions are to be fitted on the residuals of a model with main effects. If TRUE residuals of a model with environmental main effects are used, if FALSE the regressions are fitted on the original data.
outputFile	The file name of the output files, without .csv extension which is added by the function. If not NULL the prediction accuracies for training and test environments are written to separate files. If NULL the output is not written to file.
corType	type of correlation: Pearson (default) or Spearman rank sum.
partition	data.frame with columns E and partition. The column E should contain the training environments (type character); partition should be of type integer. Environments in the same fold should have the same integer value. Default is data.frame(), in which case the function uses a leave-one-environment out cross-validation. If NULL, the (inner) training sets used for cross-validation will be drawn randomly from all observations, ignoring the environment structure. In the latter case, the number of folds (nfolds) can be specified.
nfolds	Default NULL. If partition == NULL, this can be used to specify the number of folds to be used in glmnet.
alpha	Type of penalty, as in glmnet (1 = LASSO, 0 = ridge; in between = elastic net). Default is 1.
scaling	determines how the environmental variables are scaled. "train" : all data (test and training environments) are scaled using the mean and and standard deviation in the training environments. "all" : using the mean and standard deviation of all environments. "no" : No scaling.
quadratic	boolean; default FALSE. If TRUE, quadratic terms (i.e., squared indices) are added to the model.
verbose	boolean; default FALSE. If TRUE, the accuracies per environment are printed on screen.

Value

A list with the following elements:

predTrain Vector with predictions for the training set (to do: Add the factors genotype and environment; make a data.frame)

predTest Vector with predictions for the test set (to do: Add the factors genotype and environment; make a data.frame). To do: add estimated environmental main effects, not only predicted environmental main effects

mu the estimated overall (grand) mean

envInfoTrain The estimated environmental main effects, and the predicted effects, obtained when the former are regressed on the averaged indices, using penalized regression.

envInfoTest The predicted environmental main effects for the test environments, obtained from penalized regression using the estimated main effects for the training environments and the averaged indices.

parGeno data.frame containing the estimated genotypic main effects (first column) and sensitivities (subsequent columns)

testAccuracyEnv a data.frame with the accuracy (r) for each test environment

trainAccuracyEnv a data.frame with the accuracy (r) for each training environment

trainAccuracyGeno a data.frame with the accuracy (r) for each genotype, averaged over the training environments

testAccuracyGeno a `data.frame` with the accuracy (r) for each genotype, averaged over the test environments

RMSEtrain The root mean squared error on the training environments

RMSEtest The root mean squared error on the test environments

Y The name of the trait that was predicted, i.e. the column name in dat that was used

G The genotype label that was used, i.e. the argument G that was used

E The environment label that was used, i.e. the argument E that was used

indices The indices that were used, i.e. the argument indices that was used

lambdaOpt

pargeno

quadratic The quadratic option that was used

Index