Package 'METAAB'

Type Package

Title Multi Environment Trials Analysis using AMMI and BLUP

Version 1.0.2

Date 2019-02-16

Maintainer Tiago Olivoto <tiagoolivoto@gmail.com>

Description This package compute has useful functions for computing stability analysis using AMMI and BLUP methods. Several procedures, including cross-validation, prediction, and stability parameters for AMMI and BLUP models are available. Tools for personalized graphics is also provided.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

Imports dplyr,
dendextend,
heatmaply,
gplots,
ggExtra,
grid,
ggplot2,
ggrepel,
lme4,
lmerTest,
gtools,
magrittr,
methods

RoxygenNote 6.1.1

Suggests knitr,
rmarkdown,
testthat,
xlsx,
readxl,
rJava,
ReporteRs,
cowplot,
prettydoc,
kableExtra

URL https://github.com/TiagoOlivoto/METAAB

VignetteBuilder knitr

BugReports https://github.com/TiagoOlivoto/METAAB/issues

R **topics documented:**

AMMI_indexes   *AMMI-based indexes for stability and simultaneous selection*

Description

This function computes the following AMMI-based stability indexes: ASV, AMMI stability value (Purchase et al., 2000); SIPC, sums of the absolute value of the IPCA scores (Sneller et al. 1997); EV, averages of the squared eigenvector values (Sneller et al. 1997); and Za, absolute value of the relative contribution of IPCAs to the interaction (Zali et al. 2012).

Usage

```
AMMI_indexes(x, order.y = NULL)
```

Arguments

x   An object of class `WAAS.AMMI`

order.y   A vector of the same length of x used to order the response variable. Each element of the vector must be one of the "h" or "l". If "h" is used, the response variable will be ordered from maximum to minimum. If "l" is used then the response variable will be ordered from minimum to maximum.

Details

The ASV index is computed as follows:

$$ASV_i = \left[\left[\frac{r\,\lambda_1^2}{r\,\lambda_2^2} \times (\lambda_1^{0.5}\,a_{i1}t_{j1}) \right]^2 + (\lambda_2^{0.5}\,a_{i2}t_{j2})^2 \right]^{0.5}$$

where r is the number of replications included in the analysis,

The SIPC index is computed as follows:

$$SIPC_i = \sum\nolimits_{k=1}^{P} \left| |\lambda_k^{0.5}\,a_{ik}| \right|$$

where P is the number of IPCA retained via F-tests.

The EV index is computed as follows:

$$EV_i = \sum\nolimits_{k=1}^{P} a_{ik}^2 / P$$

The ZA index is computed as follows:

$$Za_i = \sum\nolimits_{k=1}^{P} \theta_k a_{ik}$$

where θ_k is the percentage sum of squares explained by the kth IPCA.

Four simultaneous selection indexes (ssi) are also computed by summation of the ranks of the ASV, SIPC, EV and Za indexes and the ranks of the mean yields (Farshadfar, 2008), which results in ssiASV, ssiSIPC, ssiEV, and ssiZa, respectively.

Value

A dataframe contaning the indexes

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

References

Purchase, J.L., H. Hatting, and C.S. van Deventer. 2000. Genotype vs environment interaction of winter wheat (Triticum aestivum L.) in South Africa: II. Stability analysis of yield performance. South African J. Plant Soil 17:101-107. doi:10.1080/02571862.2000.10634878

Sneller, C.H., L. Kilgore-Norquest, and D. Dombek. 1997. Repeatability of Yield Stability Statistics in Soybean. Crop Sci. 37:383-390. doi:10.2135/cropsci1997.0011183X003700020013x

Zali, H., E. Farshadfar, S.H. Sabaghpour, and R. Karimizadeh. 2012. Evaluation of genotype vs environment interaction in chickpea using measures of stability from AMMI model. Ann. Biol. Res. 3:3126-3136. http://eprints.icrisat.ac.in/id/eprint/7173

Examples

```
library(METAAB)
model = WAAS.AMMI(data_ge,
                  resp = c(GY, HM),
                  gen = GEN,
                  env = ENV,
                  rep = REP)
model_indexes = AMMI_indexes(model)
```

---

autoplot   *Several types of residual plots*

Description

Residual plots for a output model of class "WAAS.AMMI" and "WAASB". Six sets of plots are produced: (1) Residuals vs fitted, (2) normal Q-Q plot for the residuals, (3) scale-location plot (standardized residuals vs Fitted Values), (4) standardized residuals vs Factor-levels, (5) Histogram of raw residuals and (6) standardized residuals vs observation order. For a "WAASB" object, normal Q-Q plot for random effects may also be obtained declaring type = "re"

Usage

```
## S3 method for class 'WAAS.AMMI'
autoplot(x, conf = 0.95, labels = FALSE, theme = theme_waasb(),
        alpha = 0.2, fill.hist = "gray", col.hist = "black",
        col.point = "black", col.line = "red", col.lab.out = "red",
        size.lab.out = 2.5, size.tex.lab = 10, size.shape = 1.5,
        bins = 30, which = c(1:4), mfrow = c(2, 2), ...)


## S3 method for class 'WAASB'
```

```
autoplot(x, type = "res", conf = 0.95, out = "print",
         labels = FALSE, theme = theme_waasb(),
         alpha = 0.2, fill.hist = "gray", col.hist = "black",
         col.point = "black", col.line = "red", col.lab.out = "red",
         size.lab.out = 2.5, size.tex.lab = 10, size.shape = 1.5,
         bins = 30, which = c(1:4), mfrow = c(2, 2), ...)
```

Arguments

x	An object of class WAAS.AMMI or WAASB.
type	If type = "re", normal Q-Q plots for the random effects are obtained.
conf	Level of confidence interval to use in the Q-Q plot (0.95 by default).
out	How the output is returned. Must be one of the "print" (default) or "return".
labels	Logical argument. If TRUE labels the points outside confidence interval limits.
theme	The theme to use in the graphics. Default is theme = theme_waasb().
alpha	The transparency of confidence band in the Q-Q plot. Must be a number between 0 (opaque) and 1 (full transparency).
fill.hist	The color to fill the histogram. Default is "gray".
col.hist	The color of the border of the the histogram. Default is "black".
col.point	The color of the points in the graphic. Default is "black".
col.line	The color of the lines in the graphic. Default is "red".
col.lab.out	The color of the labels for the "outlying" points.
size.lab.out	The size of the labels for the "outlying" points.
size.tex.lab	The size of the text in axis text and labels.
size.shape	The size of the shape in the plots.
bins	The number of bins to use in the histogram. Default is 30.
which	Which graphics should be plotted. Default is which = c(1:4) that means that the first four graphics will be plotted.
mfrow	Allow creating a simple multi-paneled plot. Must be a vector of length 2. The first argument is the number of rows and the second the number of columns. Default is mfrow = c(2, 2).
...	Additional parameter for the function

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
library(METAAB)
library(ggplot2)
model = WAAS.AMMI(data_ge,
                  resp = GY,
                  gen = GEN,
                  env = ENV,
                  rep = REP)
autoplot(model$GY)
```

```
model2 = WAASB(data_ge,
               resp = c(GY, HM),
               gen = GEN,
               env = ENV,
               rep = REP)
lapply(model2, function(x){
  autoplot.WAASB(x, which = 2, out = "return")
})
```

---

corr.ci	*Nonparametric confidence interval for Pearson's correlation coefficient*

Description

Computes the half-width confidence interval for correlation coefficient based on the nonparametri method proposed by Olivoto et al. (2018).

Usage

```
corr.ci(data = NA, r = NULL, n = NULL)
```

Arguments

data	A dataset containing variables only or a symetric correlation matrix.
r	If data is not available, provide the value for correlation coefficient.
n	The sample size if data is a correlation matrix or if r is informed.

Details

The half-width confidence interval is computed according to the following equation:

$$CI_w = 0.45304^r \times 2.25152 \times n^{-0.50089}$$

where n is the sample size and r is the correlation coefficient.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

References

Olivoto, T., A.D.C. Lucio, V.Q. Souza, M. Nardino, M.I. Diel, B.G. Sari, D.. K. Krysczun, D. Meira, and C. Meier. 2018. Confidence interval width for Pearson's correlation coefficient: a Gaussian-independent estimator based on sample size and strength of association. Agron. J. 110:1-8. doi:10.2134/agronj2017.09.0566.

Examples

```
corr.ci(iris[,1:3])
```

---

corr.ss	*Sample size planning for a desired Pearson's correlation cofidence interval.*

Description

Find the required (sufficient) sample size for computing a Pearson correlation coefficient with a desired confidence interval (Olivoto et al., 2018).

Usage

```
corr.ss(r, CI)
```

Arguments

r	The magnitude of the correlation coefficient.
CI	The half-width for confidence interval at p < 0.05.

Details

The required (sufficient) sample size is computed as follows:

$$n = [CI_w/0.45304^r \times 2.25152]^{-0.50089}$$

where CI_w is desired confidence interval and r is the correlation coefficient.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

References

Olivoto, T., A.D.C. Lucio, V.Q. Souza, M. Nardino, M.I. Diel, B.G. Sari, D.. K. Krysczun, D. Meira, and C. Meier. 2018. Confidence interval width for Pearson's correlation coefficient: a Gaussian-independent estimator based on sample size and strength of association. Agron. J. 110:1-8. doi:10.2134/agronj2017.09.0566.

Examples

```
corr.ss(r = 0.60, CI = 0.1)
```

---

data_ge	*A replicate-based data of 10 genotypes cultivated in 14 environments*

Description

This dataset contain data on two variables assessed in 10 genotypes growing in in 11 environments. The experimental design was a RCBD with 3 replicates(blocks). This data provide examples for several functions of **METAAB** package.

Arguments

	The following collumns in data_ge are:
	A factor with 14 levels; each level represents one cultivation environment.
GEN	A factor with 10 levels; each level represents one genotype.
REP	A factor with 3 levels; each level represents one replication/block.
GY	A continuous variable (grain yield) observed in each plot.
HM	A continuous variable (hectoliter mass) observed in each plot.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

FAI.BLUP	*Multitrait index based on factor analysis and ideotype-design*

Description

Multitrait index based on factor analysis and ideotype-design proposed by Rocha et al. (2018).

Usage

```
FAI.BLUP(x, DI, UI, show = TRUE, SI = NULL, mineval = 1)
```

Arguments

x	An object of class WAASB
DI, UI	A vector of the same length of x to construct the desirable/undesirable ideotype. For each element of the vector, allowed values are "max", "min", "mean", or a numeric value.
show	Logical value. If TRUE the results are shown in console.
SI	An integer [0-100]. The selection intensity in percentage of the total number of genotypes.
mineval	The minimum value so that an eigenvector is retained in the factor analysis.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

References

Rocha, J.R.A.S.C.R, J.C. Machado, and P.C.S. Carneiro. 2018. Multitrait index based on factor analysis and ideotype-design: proposal and application on elephant grass breeding for bioenergy. GCB Bioenergy 10:52-60. doi: doi:10.1111/gcbb.12443.

Examples

```
multivariate = WAASB(data_ge,
                     resp = c(GY, HM),
                     gen = GEN,
                     env = ENV,
                     rep = REP)

FAI = FAI.BLUP(multivariate,
               SI = 15,
               DI = c("max", "max"),
               UI = c("min", "min"))
plot(FAI)
```

---

int.effects   *Dataset with genotype-by-environment interaction effects*

Description

This dataset contains a matrix with interaction effects of 10 genotypes cultivated in 5 environments.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

See Also

meansGxE

---

meansGxE   *A dataset with means of 10 genotypes cultivated in 5 environments*

Description

This dataset contains the means for grain yield of 10 genotypes cultivated in 5 environments. The interaction effects for this data is found in int.effects

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

---

mmat   *Make a two-way table based on categorical and numerical arguments*

Description

This function help users to easely make a two-way table from a "long format" data.

Usage

```
mmat(data, row, col, value)
```

Arguments

data	The dataset. Must contains at least two categorical columns.
row	The column of data in which the mean of each level will correspond to **one line** in the output.
col	The column of data in which the mean of each level will correspond to **one column** in the output.
value	The column of data that contains the values to fill the two-way table.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
datage = expand.grid(ENV = paste("E", 1:8, sep = ""),
                     GEN = paste("G", 1:20, sep = ""),
                     REP = paste(c("I", "II", "III")))
datage$Y = rnorm(480, 50, 20)
matrix = mmat(datage, row = GEN, col = ENV, val = Y)
matrix
```

MTSI   *Multitrait stability index*

Description

Multitrait stability index in analysis of multi-environment trials using AMMI or BLUP methods.

Usage

```
MTSI(x, index = "WAASB", show = TRUE, SI = 15, mineval = 1)
```

Arguments

x	An object of class WAASB or WAAS.AMMI.
index	If index = "WAASB" (Default) the multitrait index will be computed considering the stability of genotypes only. If index = "WAASBY" both stability and mean performance are considered. More details can be seen in WAASB and WAAS.AMMI functions.
show	If show = TRUE (Default) some results are shown in the console.
SI	An integer [0-100]. The selection intensity in percentage of the total number of genotypes.
mineval	The minimum value so that an eigenvector is retained in the factor analysis.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
library(METAAB)
# Based on stability only
MTSI_MODEL = WAASB(data_ge,
                   resp = c(GY, HM),
                   gen = GEN,
                   env = ENV,
                   rep = REP)

MTSI_index = MTSI(MTSI_MODEL)


# Based on mean performance and stability
MTSI_MODEL2 = WAASB(data_ge,
                    resp = c(GY, HM),
                    gen = GEN,
                    env = ENV,
                    rep = REP,
                    mresp = c(100, 100), #Default
                    wresp = c(65, 65))
MTSI_index2 = MTSI(MTSI_MODEL2,
                   index = "WAASBY")
plot(MTSI_index2)
```

---

partial.corr	*Partial correlation coefficient*

Description

Estimates of partial correlation coefficient based on data raw or correlation matrices.

Usage

```
partial.corr(data, n = NULL, method = "pearson")
```

Arguments

data	The data, must be a dataframe or a symetric correlation matrix.
n	The number of subjects used if data is a correlation matrix.
method	If data is a correlation matrix, then codemethod is the method for computing the correlation; one of the "pearson" (default), "kendall", or "spearman"

Value

linear.mat	The matrix of linear correlation.
partial.mat	The matrix of partial correlations.
results	Hypotesis testing for each pairwise comparation
call	The function call

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
partial.corr(iris[,1:4])
```

---

performs_ammi	*Performs Additive Main effects and Multiplicative Interaction*

Description

Compute the Additive Main effects and Multiplicative interaction. This is a helper function for other procedures performed in the WAASB package such as WAAS.AMMI, validation.AMMI, validation.AMMIF, and predict.WAAS.AMMI.

Usage

```
performs_ammi(ENV, GEN, REP, Y)
```

Arguments

ENV	The name of the collum that contains the levels of the environments
GEN	The name of the collum that contains the levels of the genotypes
REP	The name of the collum that contains the levels of the replications/blocks
Y	The response variable

Value

ANOVA	The analysis of variance for the AMMI model.
analysis	The principal component analysis
means	means of genotype vs environment
biplot	escores for genotypes and environments in all the possible axes.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
library(METAAB)
ammi_model = with(data_ge, performs_ammi(ENV, GEN, REP, GY))
```

---

plot.blup   *Plot the BLUPs for genotypes*

Description

Plot the predicted BLUP of the genotypes.

Usage

```
## S3 method for class 'blup'
plot(x, prob = 0.05, export = FALSE, file.type = "pdf",
      file.name = NULL, theme = theme_waasb(), width = 6,
      height = 6, size.err.bar = 0.5, size.shape = 3.5,
      size.tex.lab = 12, height.err.bar = 0.3, x.lim = NULL,
      x.breaks = waiver(), col.shape = c("blue", "red"),
      y.lab = "Genotypes", x.lab = "Predicted Grain Yield",
      resolution = 300, ...)
```

Arguments

x	The WAASB object
prob	The probability error for constructing confidence interval.
export	Export (or not) the plot. Default is TRUE.
file.type	If export = TRUE, define the type of file to be exported. Default is pdf, Graphic can also be exported in *.tiff format by declaring file.type = "tiff".
file.name	The name of the file for exportation, default is NULL, i.e. the files are automatically named.
theme	The graphical theme of the plot. Default is 'theme = theme_waasb()'. Please, see '?WAASB::theme_waasb'. An own theme can be applied using the arguments: 'theme = theme_waasb() + theme(some stuff here)'. For more details, please, see ' ?ggplot2::theme'
width	The width "inch" of the plot. Default is 6.
height	The height "inch" of the plot. Default is 6.
size.err.bar	The size of the error bar for the plot. Default is 0.5.
size.shape	The size of the shape (both for genotypes and environments). Default is 3.5.
size.tex.lab	The size of the text in axis text and labels.
height.err.bar	The height for error bar. Default is 0.3.
x.lim	The range of x-axis. Default is NULL (maximum and minimum values of the data set). New arguments can be inserted as x.lim = c(x.min, x.max).

x.breaks	The breaks to be plotted in the x-axis. Default is `authomatic breaks`. New arguments can be inserted as `x.breaks = c(breaks)`
col.shape	A vector of length 2 that contains the color of shapes for genotypes above and below of the mean, respectively. Default is `c("blue", "red")`.
x.lab	The label of the x-axis in the plot. Default is `"Predicted Grain Yield"`.
y.lab	The label of the y-axis in the plot. Default is `"Genotypes"`.
resolution	The resolution of the plot. Parameter valid if `file.type = "tiff"` is used. Default is `300` (300 dpi)
...	Other arguments of the function

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

See Also

plot.scores, plot.WAASBY

Examples

```
library(METAAB)
BLUP = WAASB(data_ge,
             resp = c(GY, HM),
             gen = GEN,
             env = ENV,
             rep = REP)
plot.blup(BLUP$GY)
```

---

plot.eigen   *Plot the eigenvalues*

Description

Plot the eigenvalues for from singular value decomposition of BLUP interaction effects matrix.

Usage

```
## S3 method for class 'eigen'
plot(x, export = FALSE, theme = theme_waasb(),
     file.type = "pdf", file.name = NULL, width = 6, height = 6,
     size.shape = 3.5, size.line = 1, size.tex.lab = 12,
     y.lab = "Eigenvalue",  y2.lab = "Accumulated variance",
     x.lab = "Number of multiplicative terms", resolution = 300, ...)
```

Arguments

x	The WAASB object
export	Export (or not) the plot. Default is TRUE.
theme	The graphical theme of the plot. Default is 'theme = theme_waasb()'. Please, see '?WAASB::theme_waasb'. An own theme can be applied using the arguments: 'theme = theme_waasb() + theme(some stuff here)'. For more details, please, see '?ggplot2::theme'
file.type	If export = TRUE, define the type of file to be exported. Default is pdf, Graphic can also be exported in *.tiff format by declaring file.type = "tiff".
file.name	The name of the file for exportation, default is NULL, i.e. the files are automatically named.
width	The width "inch" of the plot. Default is 6.
height	The height "inch" of the plot. Default is 6.
size.shape	The size of the shape. Default is 3.5.
size.line	The size of the line. Default is 1.
size.tex.lab	The size of the text in axis text and labels.
y.lab	The label of the y-axis in the plot. Default is "Eigenvalue".
y2.lab	The label of the second y-axis in the plot. Default is "Accumulated variance".
x.lab	The label of the x-axis in the plot. Default is "Number of multiplicative terms".
resolution	The resolution of the plot. Parameter valid if file.type = "tiff" is used. Default is 300 (300 dpi)
...	Other arguments of the function

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

See Also

plot.scores, plot.WAASBY

Examples

```
library(METAAB)
BLUP = WAASB(data_ge,
             resp = c(GY, HM),
             gen = GEN,
             env = ENV,
             rep = REP)
plot.eigen(BLUP$GY)
```

`plot.FAI.BLUP` *Multitrait index based on factor analysis and ideotype-design*

Description

Plot the multitrait index based on factor analysis and ideotype-design proposed by Rocha et al. (2018).

Usage

```
## S3 method for class 'FAI.BLUP'
plot(x, ideotype = 1, SI = 15, radar = TRUE,
     size.point = 2, col.sel = "red",
     col.nonsel = "black", ...)
```

Arguments

x	An object of class WAASB
ideotype	The ideotype to be plotted. Default is 1.
SI	An integer [0-100]. The selection intensity in percentage of the total number of genotypes.
radar	Logical argument. If true (default) a radar plot is generated after using `coord_polar()`.
size.point	The size of the point in graphic.
col.sel	The colour for selected genotypes.
col.nonsel	The colour for nonselected genotypes.
...	Other arguments of the function.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

References

Rocha, J.R.A.S.C.R, J.C. Machado, and P.C.S. Carneiro. 2018. Multitrait index based on factor analysis and ideotype-design: proposal and application on elephant grass breeding for bioenergy. GCB Bioenergy 10:52-60. doi: doi:10.1111/gcbb.12443.

Examples

```
multivariate = WAASB(data_ge,
                     resp = c(GY, HM),
                     gen = GEN,
                     env = ENV,
                     rep = REP)

FAI = FAI.BLUP(multivariate,
               SI = 15,
               DI = c("max", "max"),
               UI = c("min", "min"))
plot(FAI)
```

---

plot.MTSI   *Plot the multitrait stability index*

Description

Plot the multitrait stability index using ggplot-generated graphics. .

Usage

```
## S3 method for class 'MTSI'
plot(x,  SI = 15, radar = TRUE, size.point = 2,
     col.sel = "red", col.nonsel = "black", ...)
```

Arguments

x	An object of class MTSI
SI	An integer [0-100]. The selection intensity in percentage of the total number of genotypes.
radar	Logical argument. If true (default) a radar plot is generated after using coord_polar().
size.point	The size of the point in graphic.
col.sel	The colour for selected genotypes.
col.nonsel	The colour for nonselected genotypes.
...	Other arguments of the function.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
library(METAAB)
MTSI_MODEL = WAASB(data_ge,
                   resp = c(GY, HM),
                   gen = GEN,
                   env = ENV,
                   rep = REP)

MTSI_index = MTSI(MTSI_MODEL)
plot(MTSI_index)
```

plot.ratios   *Plot heat maps with genotype ranking*

Description

Plot heat maps with genotype ranking in two ways.

Usage

```
## S3 method for class 'WAASBYratio'
plot(x, type, export = FALSE, file.type = "pdf",
     file.name = NULL, width = 6, height = 5, size.lab = 1,
     margins = c(5, 4),  y.lab = NULL, x.lab = NULL,
     key.lab = "Genotype ranking", resolution = 300, ...)

## S3 method for class 'WAASratio.AMMI'
plot(x, type, export = FALSE, file.type = "pdf",
     file.name = NULL, width = 6, height = 5, size.lab = 1,
     margins = c(5, 4),  y.lab = NULL, x.lab = NULL,
     key.lab = "Genotype ranking", resolution = 300, ...)
```

Arguments

x	The `WAASYratio` object
type	`1 = Heat map Ranks`: this graphic shows the genotype ranking considering the WAAS estimated with different numbers of Principal Components; `2 = Heat map WAASY-GY ratio`: this graphic shows the genotype ranking considering the different combinations in the WAAS/GY ratio.
export	Export (or not) the plot. Default is `TRUE`.
file.type	If `export = TRUE` define the type of file to be exported. Default is `pdf`, Graphic can also be exported in `*.tiff` format by declaring `file.type = "tiff"`.
file.name	The name of the file for exportation, default is `NULL`, i.e. the files are automatically named.
width	The width "inch" of the plot. Default is 8.
height	The height "inch" of the plot. Default is 7.
size.lab	The label size of the plot. It is suggested attribute 1
margins	Numeric vector of length 2 containing the margins for column and row names, respectively. Default is `c(5, 4)`.
y.lab	The label of y axis. Default is "Genotypes".
x.lab	The label of x axis. Default is "Number of axes".
key.lab	The label of color key. Default is "Genotype ranking".
resolution	Valid parameter if `file.type = "tiff"`. Define the resolution of the plot. Default is "300".
...	Other arguments of the function

Details

The first type of heatmap shows the genotype ranking depending on the number of principal component axis used for estimating the WAASB index. An euclidian distance-based dendrogram is used for grouping the genotype ranking for both genotypes and principal component axis. The second type of heatmap shows the genotype ranking depending on the WAASB/GY ratio. The ranks obtained with a ratio of 100/0 considers exclusively the stability for the genotype ranking. On the other hand, a ratio of 0/100 considers exclusively the productivity for the genotype ranking. Four clusters are estimated (1) unproductive and unstable genotypes; (2) productive, but unstable genotypes; (3) stable, but unproductive genotypes; and (4), productive and stable genotypes.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
library(METAAB)
#wratio = WAASBYratio(data_ge,
#                     resp = GY,
#                     gen = GEN,
#                     env = ENV,
#                     rep = REP,
#                     increment = 10)
#plot(wratio, type = 1)

#wratio2 = WAASratio.AMMI(data_ge,
#                         resp = GY,
#                         gen = GEN,
#                         env = ENV,
#                         rep = REP,
#                         increment = 5)
#plot(wratio2, type = 2)
```

---

plot.scores	*Plot scores in different graphical interpretations*

Description

Plot scores of genotypes and environments in different graphics. 1 = PC1 x PC2, 2 = GY x PC1, 3 = GY x WAASB, and 4 = Nominal yield x EPCA1.

Usage

```
## S3 method for class 'scores'
plot(x, type = 1, polygon = FALSE, file.type = "pdf",
     export = FALSE, file.name = NULL, theme = theme_waasb(),
     axis.expand = 1.1, width = 8, height = 7, x.lim = NULL,
     x.breaks = waiver(), x.lab = NULL, y.lab = NULL, y.lim = NULL,
     y.breaks = waiver(), shape.gen = 21, shape.env = 23,
     size.shape = 2.2, size.bor.tick = 0.3, size.tex.lab = 12,
     size.tex.pa = 3.5, size.line = 0.5, size.segm.line = 0.5,
     leg.lab = c("Gen", "Env"), line.type = "solid", line.alpha = 0.9,
```

```
        col.line = "gray", col.gen = "blue", col.env = "darkgreen",
        col.alpha = 0.9, col.segm.gen = "transparent",
        col.segm.env = "darkgreen", resolution = 300, ...)
```

Arguments

x	The object WAASB or WAAS.AMMI
type	Three types of graphics can be generated: 1 = PC1 x PC2, default, to make inferences related to the interaction effects; 2 = GY x PC1 to make inferences related to stability and productivity; 3 = GY x WAASB, and 4 = Nominal yield x Environment PC1.
polygon	Logical argument. If TRUE, a polygon is drawn when type 1.
file.type	The type of file to be exported. Valid parameter if export = T|TRUE. Default is "pdf". The graphic can also be exported in *.tiff format by declaring file.type = "tiff".
export	Export (or not) the plot. Default is FALSE.
file.name	The name of the file for exportation, default is NULL, i.e. the files are automatically named.
theme	The graphical theme of the plot. Default is 'theme = theme_waasb()'. Please, see '?WAASB::theme_waasb'. An own theme can be applied using the arguments: 'theme = theme_waasb() + theme(some stuff here)'. For more details, please, see '?ggplot2::theme'
axis.expand	Multiplication factor to expand the axis limits by to enable fitting of labels. Default is 1.1.
width	The width "inch" of the plot. Default is 8.
height	The height "inch" of the plot. Default is 7.
x.lim	The range of x-axis. Default is NULL (maximum and minimum values of the data set). New arguments can be inserted as x.lim = c(x.min, x.max).
x.breaks	The breaks to be plotted in the x-axis. Default is authomatic breaks. New arguments can be inserted as x.breaks = c(breaks)
x.lab	The label of x-axis. Each plot has a default value. New arguments can be inserted as x.lab = "my label".
y.lab	The label of y-axis. Each plot has a default value. New arguments can be inserted as y.lab = "my label".
y.lim	The range of x-axis. Default is NULL. The same arguments than x.lim can be used.
y.breaks	The breaks to be plotted in the x-axis. Default is authomatic breaks. The same arguments than x.breaks can be used.
shape.gen	The shape for genotype indication in the biplot. Default is 21 (circle). Values must be between 21-25: 21 (circle), 22 (square), 23 (diamond), 24 (up triangle), and 25 (low triangle).
shape.env	The shape for environment indication in the biplot. Default is 23 (diamond). The same arguments than "shape.gen".
size.shape	The size of the shape (both for genotypes and environments). Default is 2.2.
size.bor.tick	The size of tick of shape. Default is 0.3. The size of the shape will be size.shape + size.bor.tick
size.tex.lab	The size of the text in the axes text and labels. Default is 12.

size.tex.pa	The size of the text of the plot area. Default is 3.5.
size.line	The size of the line that indicate the means in the biplot. Default is 0.5.
size.segm.line	The size of the segment that start in the origin of the biplot and end in the scores values. Default is 0.5.
leg.lab	The labs of legend. Default is Gen and Env.
line.type	The type of the line that indicate the means in the biplot. Default is "solid". Other values that can be attributed are: "blank", no lines in the biplot, "dashed", "dotted", "dotda
line.alpha	The alpha value that combine the line with the background to create the appearance of partial or full transparency. Default is 0.4. Values must be between "0" (full transparency) to "1" (full color).
col.line	The color of the line that indicate the means in the biplot. Default is "gray"
col.gen	The shape color for genotypes. Must be one value or a vector of colours with the same length of the number of genotypes. Default is "blue". Other values can be attributed. For example, "transparent", will make a plot with only an outline around the shape area.
col.env	The shape color for environments. Default is "red". The same usability than "col.gen".
col.alpha	The alpha value for the color. Default is 0.9. Values must be between 0 (full transparency) to 1 (full color).
col.segm.gen	The color of segment for genotypes.Default is "transparent". Parameter valid for type = 1 and type = 2 graphics. This segment start in the origin of the biplot and end in the scores values.
col.segm.env	The color of segment for environments. Default is "gray50". The same usability than "col.segm.gen"
resolution	The resolution of the plot. Parameter valid if file.type = "tiff" is used. Default is 300 (300 dpi)
...	Other arguments of the function

Details

The plots type 1 and 2 have the same interpretation than those used in traditional-usage AMMI analysis (well know as AMMI2 and AMMI1, respectively). In the plot type 3, the scores of both genotypes and environments are plotted considering the response variable and the WAASB (stability index that considers all significant principal component axis of traditional AMMI models or all principal component axis estimated with BLUP-interaction effects. Plot type 4 may be used to better understand the well known "which-won-where" pattern, facilitating the recommendation of appropriate genotypes targeted for specific environments, thus allowing the exploitation of narrow adaptations.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

See Also

[plot.eigen](), [plot.validation.AMMIF]()

Examples

```
library(METAAB)
library(ggplot2)
scores = WAASB(data_ge,
               resp = GY,
               gen = GEN,
               env = ENV,
               rep = REP)
# PC1 x PC2
plot.scores(scores$GY,
            type = 1,
            polygon = TRUE)

# GY x PC1
plot.scores(scores$GY,
            type = 2,
            col.env = "olivedrab",
            col.gen = "orange2",
            x.lab = "My own x label")

# GY x WAASB
plot.scores(scores$GY,
            type = 3,
            size.tex.pa = 2,
            size.tex.lab = 16)
```

---

plot.validation.AMMIF   *Plot the RMSPD of all AMMI-family tested models*

Description

Boxplot showing the Root Means Square Prediction Difference of all AMMI family models obtained in a cross-validation procedure.

Usage

```
## S3 method for class 'validation.AMMIF'
plot(x, violin = FALSE, export = FALSE,  x.lab = NULL,
     y.lab = NULL, size.tex.lab = 12, file.type = "pdf",
     file.name = NULL, theme = theme_waasb(),width = 6,
     height = 6, resolution = 300, col.violin = "gray90",
     col.boxplot = "gray70", width.boxplot = 0.2,
     x.lim = NULL, x.breaks = waiver(), ...)
```

Arguments

x	An object of class `validation.AMMIF`.
violin	Define if a violine plot is used with boxplot. Default is "TRUE"
export	Export (or not) the plot. Default is `T`.
x.lab	The label of x-axis. New arguments can be inserted as `x.lab = "my x label"`.

y.lab	The label of y-axis. New arguments can be inserted as y.lab = "my y label".
size.tex.lab	The size of the text in axis text and labels.
file.type	The type of file to be exported. Default is pdf, Graphic can also be exported in *.tiff format by declaring file.type = "tiff".
file.name	The name of the file for exportation, default is NULL, i.e. the files are automatically named.
theme	The graphical theme of the plot. Default is 'theme = theme_waasb()'. Please, see '?WAASB::theme_waasb'. An own theme can be applied using the arguments: 'theme = theme_waasb() + theme(some stuff here)'. For more details, please, see '?ggplot2::theme'
width	The width "inch" of the plot. Default is 6.
height	The height "inch" of the plot. Default is 6.
resolution	The resolution of the plot. Parameter valid if file.type = "tiff" is used. Default is 300 (300 dpi)
col.violin	Parameter valid if violin = T. Define the color of the violin plot. Default is "gray90.
col.boxplot	Define the color for boxplot. Default is "gray70".
width.boxplot	The width of boxplots. Default is 0.2.
x.lim	The range of x-axis. Default is NULL (maximum and minimum values of the data set). New arguments can be inserted as x.lim = c(x.min, x.max).
x.breaks	The breaks to be plotted in the x-axis. Default is authomatic breaks. New arguments can be inserted as x.breaks = c(breaks)
...	Other arguments of the function

Details

Five statistics are shown in this type of plot. The lower and upper hinges correspond to the first and third quartiles (the 25th and 75th percentiles). The upper whisker extends from the hinge to the largest value no further than 1.5 * IQR from the hinge (where IQR is the inter-quartile range). The lower whisker extends from the hinge to the smallest value at most 1.5 * IQR of the hinge. Data beyond the end of the whiskers are considered outlying points.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

See Also

[plot.scores](), [plot.WAASBY]()

Examples

```
validation <- validation.AMMIF(data_ge,
                              resp = GY,
                              gen = GEN,
                              env = ENV,
                              rep = REP,
                              nboot = 5,
                              nrepval = 2)
plot(validation)
```

plot.WAASBY   *Plot WAASBY values for genotype ranking*

Description

Plot heat maps with genotype ranking in two ways.

Usage

```
## S3 method for class 'WAASBY'
plot(x, export = F , file.type = "pdf", file.name = NULL,
    theme = theme_waasb(), width = 6, height = 6, size.shape = 3.5,
    size.tex.lab = 12, col.shape = c("blue", "red"), x.lab = "WAASBY",
    y.lab = "Genotypes", x.breaks = waiver(), resolution = 300, ...)
```

Arguments

x	The WAASBY object
export	Export (or not) the plot. Default is T.
file.type	The type of file to be exported. Default is pdf, Graphic can also be exported in *.tiff format by declaring file.type = "tiff".
file.name	The name of the file for exportation, default is NULL, i.e. the files are automatically named.
theme	The graphical theme of the plot. Default is 'theme = theme_waasb()'. Please, see '?WAASB::theme_waasb'. An own theme can be applied using the arguments: 'theme = theme_waasb() + theme(some stuff here)'. For more details, please, see ' ?ggplot2::theme'
width	The width "inch" of the plot. Default is 8.
height	The height "inch" of the plot. Default is 7.
size.shape	The size of the shape in the plot. Default is 3.5.
size.tex.lab	The size of the text in axis text and labels.
col.shape	A vector of length 2 that contains the color of shapes for genotypes above and below of the mean, respectively. Default is c("blue", "red").
x.lab	The label of the x axis in the plot. Default is "WAASBY".
y.lab	The label of the y axis in the plot. Default is "Genotypes".
x.breaks	The breaks to be plotted in the x-axis. Default is authmatic breaks. New arguments can be inserted as x.breaks = c(breaks)
resolution	The resolution of the plot. Parameter valid if file.type = "tiff" is used. Default is 300 (300 dpi)
...	Other arguments of the function

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

See Also

plot.scores

Examples

```
library(METAAB)
library(ggplot2)
waasby = WAASB(data_ge,
               resp = GY,
               gen = GEN,
               env = ENV,
               rep = REP)
waasby2 = WAAS.AMMI(data_ge,
                    resp = GY,
                    gen = GEN,
                    env = ENV,
                    rep = REP)
plot.WAASBY(waasby$GY)
plot.WAASBY(waasby2$GY) +
            theme_gray() +
            theme(legend.position = "bottom",
                  legend.background = element_blank(),
                  legend.title = element_blank(),
                  legend.direction = "horizontal")
```

---

predict.WAAS.AMMI	*Predict the means of a WAAS.AMMI object*

Description

Predict the means of a WAAS.AMMI object considering a specific number of axis.

Usage

```
## S3 method for class 'WAAS.AMMI'
predict(object, naxis, ...)
```

Arguments

object	An object of class WAAS.AMMI
naxis	The the number of axis to be use in the prediction. If object has more than one variable, then naxis must be a vector.
...	Additional parameter for the function

Details

This function is used to predict the response variable of a two-way table (for examples the yielding of the i-th genotype in the j-th environment) based on AMMI model. This prediction is based on the number of multiplicative terms used. If naxis = 0, only the main effects (AMMI0) are used. In this case, the predicted mean will be the predicted value from OLS estimation. If naxis = 1 the AMMI1 (with one multiplicative term) is used for predicting the response variable. If naxis = min(gen-1;env-1), the AMMIF is fitted and the predicted value will be the cell mean, i.e. the mean of R-replicates of the i-th genotype in the j-th environment. The number of axis to be used must be carrefully chosen. Precures based on Postdictive sucess (such as Gollobs's d.f.) or Predictive sucess (such as cross-validation) should be used to do this. This package provide both.

WAAS.AMMI function compute traditional AMMI analysis showing the number of significant axis. On the other hand, validation.AMMIF function provide a cross-validation, estimating the RMSE of all AMMI-family models, based on resampling procedure.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
library(METAAB)
model = WAAS.AMMI(data_ge,
                  resp = c(GY, HM),
                  gen = GEN,
                  env = ENV,
                  rep = REP)
# Predict GY with 3 IPCA and HM with 1 IPCA
predict = predict.WAAS.AMMI(model, naxis = c(3, 1))
```

---

rbind_fill	*Helper function for combining data.frames by row, filling missing values.*

Description

Helper function similar to plyr::rbind.fill that combines data.frames by row and fills with . missing values

Usage

```
rbind_fill(...)
```

Arguments

...	Input dataframes.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
df1 = data.frame(v1 = c(1, 2), v2 = c(2, 3))
df2 = data.frame(v3 = c(4, 5))
rbind_fill(df1, df2)
```

---

resca *Rescale a continuous vector to have specified minimum and maximum values*

Description

Helper function used in the WAASB package. It rescales a continuous vector to have specified minimum and maximum values. Missing values are not allowed.

Usage

```
resca(values, new_min = 0, new_max = 100)
```

Arguments

values	continuous vector of values to manipulate.
new_min	The minimum value of the new scale. Default is 0.
new_max	The maximum value of the new scale. Default is 100

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
resca(1:10)
resca(c(20, 30, 40), new_min = 5, new_max = 10)
```

Resende_indexes *Stability indexes proposed by Resende based on a mixed-effect model.*

Description

This function computes the following indexes proposed by Resende (2007): the harmonic mean of genotypic values (HMGV), the relative performance of the genotypic values (RPGV) and the harmonic mean of the relative performance of genotypic values (HMRPGV).

Usage

```
Resende_indexes(x)
```

Arguments

x	An object of class WAASB

Details

The indexes computed with this function have been used to select genotypes with stability performance in a mixed-effect model framework. Some examples are in Alves et al (2018), Azevedo Peixoto et al. (2018), Dias et al. (2018) and Colombari Filho et al. (2013).

The HMGV index is computed as

$$HMGV_i = \frac{1}{E} \sum_{j=1}^{E} \frac{1}{Gv_{ij}}$$

where E is the number of environments included in the analysis, Gv_{ij} is the genotypic value (BLUP) for the ith genotype in the jth environment.

The RPGV index is computed as

$$RPGV_i = \frac{1}{e} \sum_{j=1}^{e} BLUP_{ij}/\mu_j$$

The HMRPGV index is computed as

$$HMRPGV_i = \frac{1}{E} \sum_{j=1}^{E} \frac{1}{Gv_{ij}/\mu_j}$$

Value

A dataframe contaning the indexes

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

References

Alves, R.S., L. de Azevedo Peixoto, P.E. Teodoro, L.A. Silva, E.V. Rodrigues, M.D.V. de Resende, B.G. Laviola, and L.L. Bhering. 2018. Selection of Jatropha curcas families based on temporal stability and adaptability of genetic values. Ind. Crops Prod. 119:290-293. doi:10.1016/J.INDCROP.2018.04.029.

Colombari Filho, J.M., M.D.V. de Resende, O.P. de Morais, A.P. de Castro, E.P. Guimaraes, J.A. Pereira, M.M. Utumi, and F. Breseghello. 2013. Upland rice breeding in Brazil: a simultaneous genotypic evaluation of stability, adaptability and grain yield. Euphytica 192:117-129. doi:10.1007/s10681-013-0922-2

Dias, P.C., A. Xavier, M.D.V. de Resende, M.H.P. Barbosa, F.A. Biernaski, R.A. Estopa. 2018. Genetic evaluation of Pinus taeda clones from somatic embryogenesis and their genotype x environment interaction. Crop Breed. Appl. Biotechnol. 18:55-64. doi:10.1590/1984-70332018v18n1a8

Azevedo Peixoto, L. de, P.E. Teodoro, L.A. Silva, E.V. Rodrigues, B.G. Laviola, and L.L. Bhering. 2018. Jatropha half-sib family selection with high adaptability and genotypic stability. PLoS One 13:e0199880. doi:10.1371/journal.pone.0199880.

Resende MDV (2007) Matematica e estatistica na analise de experimentos e no melhoramento genetico. Embrapa Florestas, Colombo

Examples

```
library(METAAB)
res_ind = WAASB(data_ge,
                resp = c(GY, HM),
                gen = GEN,
                env = ENV,
                rep = REP)
model_indexes = Resende_indexes(res_ind)
```

---

summary.MTSI   *Summary a MTSI object*

Description

Summary a MTSI object in two ways. By default, the results are shown in the R console. The results can also be exported to the directory.

Usage

```
## S3 method for class 'MTSI'
summary(object, export = FALSE, file.name = NULL, digits = 4, ...)
```

Arguments

object	The MTSI object
export	A logical argument. If TRUE|T, a *.txt file is exported to the working directory
file.name	The name of the file if export = TRUE
digits	The significant digits to be shown.
...	Other arguments of the function

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
library(METAAB)
# Based on stability only
MTSI_MODEL = WAASB(data_ge,
                   resp = c(GY, HM),
                   gen = GEN,
                   env = ENV,
                   rep = REP)

MTSI_index = MTSI(MTSI_MODEL)
summary(MTSI_index)
summary(MTSI_index,
        export = TRUE,
        file.name = "my results")
```

---

summary.WAAS.AMMI   *Summary a WAAS.AMMI object*

Description

Summary the `WAAS.AMMI` object in two ways. By default, the results are shown in the R console. The results can also be exported to the directory.

Usage

```
## S3 method for class 'WAAS.AMMI'
summary(object, export = FALSE, file.name = NULL, digits = 4, ...)
```

Arguments

`object`	The `WAAS.AMMI` object
`export`	A logical argument. If `TRUE|T`, a *.txt file is exported to the working directory
`file.name`	The name of the file if `export = TRUE`
`digits`	The significant digits to be shown.
`...`	Other arguments of the function

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
library(METAAB)
model = WAAS.AMMI(data_ge,
                  resp = c(GY, HM),
                  gen = GEN,
                  env = ENV,
                  rep = REP)
summary(model)
```

---

summary.WAASB   *Summary a WAASB object*

Description

Summary the `WAASB` object in two ways. By default, the results are shown in the R console. The results can also be exported to the directory.

Usage

```
## S3 method for class 'WAASB'
summary(object, export = FALSE, blup = FALSE,
        file.name = NULL, digits = 4, ...)
```

Arguments

object	The WAASB object
export	A logical argument. If TRUE|T, a *.txt file is exported to the working directory
blup	A logical argument. If TRUE|T, the blups are shown.
file.name	The name of the file if export = TRUE
digits	The significant digits to be shown.
...	Other arguments of the function

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
library(METAAB)
model = WAASB(data_ge,
              resp = c(GY, HM),
              gen = GEN,
              env = ENV,
              rep = REP)
summary(model)
```

---

theme_waasb	*A personalized theme for the WAASB ggplot2-based graphics*

Description

The function theme_waasb() is a personalized theme for the WAASB ggplot2-based graphics. An own theme can be applied in all graphics by using the following arguments in the plot. functions of the package: theme = theme_waasb() + theme(some stuff here...). See ?ggplot2::theme for more details.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

---

validation.AMMI	*Cross-validation for estimation of AMMI model*

Description

Cross-validation for estimation of AMMI models

Usage

```
validation.AMMI(data, resp, gen, env, rep, design = "RCBD",
                nboot, nrepval, naxis, progbar = TRUE)
```

Arguments

data	The dataset containing the columns related to Environments, Genotypes, replication/block and response variable(s).
resp	The response variable, for example resp = "RG".
gen	The name of the column that contains the levels of the genotypes.
env	The name of the column that contains the levels of the environments.
rep	The name of the column that contains the levels of the replications/blocks.
design	The experimental desig to be considered. Default is RCBD (Randomized complete Block Design). For Completely Randomized Designs inform design = "CRD".
nboot	The number of resamples to be used in the cross-validation
nrepval	The number of replicates (r) from total number of replicates (R) to be used in the modeling dataset. Only one replicate is used as validating data each step, so, Nrepval must be equal R-1
naxis	The number of axis to be considered for estimation of GE effects.
progbar	A logical argument to define if a progress bar is shown. Default is TRUE.

Details

For each iteration, the original dataset is split into two datasets: modeling and validating data. The dataset "modeling" has all combinations (genotype x environment) with the number of replications informed in nrepval. The dataset "validating" has one replication. The splitting of the dataset into modeling and validating data depends on the design informed. For Completely Randomized Block Design (default), compltely blocks are selected within environments. The remained block serves validation data. If design = "RCD" is informed, completely randomly samples are made for each genotype-by-environment combination. The estimated values (depending on NAXIS informed) are compared with the "validating" data. the Root Means Square error is computed. At the end of boots, a list is returned with the following values.

Value

RMSE	A vector with Nboot-estimates of the root mean squared error estimated with the difference between predicted and validating data.
RSMEmean	The mean of RMSE estimates.
Estimated	A data frame that contain the values (predicted, observed, validation) of the last loop.
Modeling	The dataset used as modeling data in the last loop.
Testing	The dataset used as testing data in the last loop.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

See Also

validation.AMMIF

Examples

```
library(METAAB)
model = validation.AMMI(data_ge,
                        resp = GY,
                        gen = GEN,
                        env = ENV,
                        rep = REP,
                        nboot = 5,
                        nrepval = 2,
                        naxis = 2)
```

---

validation.AMMIF	*Cross-validation for estimation of all AMMI-family models*

Description

Cross-validation for estimation of all AMMI-family models

Usage

```
validation.AMMIF(data, resp, gen, env, rep, design = "RCBD",
                 nboot, nrepval, progbar = TRUE)
```

Arguments

data	The dataset containing the columns related to Environments, Genotypes, replication/block and response variable(s).
resp	The response variable, for example resp = GY.
gen	The name of the column that contains the levels of the genotypes.
env	The name of the column that contains the levels of the environments.
rep	The name of the column that contains the levels of the replications/blocks.
design	The experimental desig to be considered. Default is RCBD (Randomized complete Block Design). For Completely Randomized Designs inform design = "CRD".
nboot	The number of resamples to be used in the cross-validation
nrepval	The number of replicates (r) from total number of replicates (R) to be used in the modeling dataset. Only one replicate is used as validating data each step, so, Nrepval must be equal R-1
progbar	A logical argument to define if a progress bar is shown. Default is TRUE.

Details

This function provides a complete cross-validation of replicate-based data using AMMI-family models. By default, the first validation is carried out considering the AMMIF (all possible axis used). Considering this model, the original dataset is split up into two datasets: training set and validation set. The "training" set has all combinations (genotype x environment) with the number of replications informed in nrepval. The dataset "validation" set has the remaining replication. The splitting of the dataset into modeling and validating data depends on the design informed. For

Completely Randomized Block Design (default), completely blocks are selected within environ-
ments. The remained block serves validation data. If design = "RCD" is informed, completely
randomly samples are made for each genotype-by-environment combination. The estimated values
(depending on the naxis informed) are compared with the "validation" data. the Root Mean Square
Prediction Difference (RMSPD) is computed. At the end of boots, a list is returned.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

See Also

validation.AMMI

Examples

```
library(METAAB)
model = validation.AMMIF(data_ge,
                         resp = GY,
                         gen = GEN,
                         env = ENV,
                         rep = REP,
                         nboot = 2,
                         nrepval = 2)
```

---

validation.blup   *Cross-validation for blup prediction*

Description

Cross-validation for blup prediction.

Usage

```
validation.blup(data, resp, gen, env, rep,
                nboot, nrepval, progbar = TRUE)
```

Arguments

data	The dataset containing the columns related to Environments, Genotypes, repli-cation/block and response variable(s).
resp	The response variable, for example resp = GY.
gen	The name of the column that contains the levels of the genotypes.
env	The name of the column that contains the levels of the environments.
rep	The name of the column that contains the levels of the replications/blocks.
nboot	The number of resamples to be used in the cross-validation
nrepval	The number of replicates (r) from total number of replicates (R) to be used in the modeling dataset. Only one replicate is used as validating data each step, so, Nrepval must be equal R-1
progbar	A logical argument to define if a progress bar is shown. Default is TRUE.

Details

This function provides a cross-validation procedure for mixed models using replicate-based data. By default, complete blocks are randomly selected within each evironment. In each iteration, the original dataset is split up into two datasets: training and validation data. The "training" set has all combinations (genotype x environment) with the number of replications informed in `nrepval`. The "validation" set has the remaining replication. The estimated values are compared with the "validation" data and the Root Means Square Prediction Difference is computed. At the end of boots, a list is returned.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

See Also

[plot.scores](), [plot.WAASBY]()

Examples

```
library(METAAB)
model = validation.blup(data_ge,
                        resp = GY,
                        gen = GEN,
                        env = ENV,
                        rep = REP,
                        nboot = 5,
                        nrepval = 2)
```

WAAS.AMMI	*Weighted Average of Absolute Scores for AMMI analysis*

Description

Compute the Weighted Average of Absolute Scores for AMMI analysis.

Usage

```
WAAS.AMMI(data, resp, gen, env, rep, mresp = NULL,
          wresp = NULL, prob = 0.05, naxis = NULL,
          verbose = TRUE)
```

Arguments

data	The dataset containing the columns related to Environments, Genotypes, replication/block and response variable(s).
resp	The response variable(s). To analyze multiple variables in a single procedure a vector of variables may be used. For example `resp = c(var1, var2, var3`.
gen	The name of the column that contains the levels of the genotypes.
env	The name of the column that contains the levels of the environments.
rep	The name of the column that contains the levels of the replications/blocks.

mresp	A numeric vector of the same length of resp. The mresp will be the new maximum value after rescaling. By default, all variables in resp are rescaled so that de maximum value is 100 and the minimum value is 0.
wresp	The weight for the response variable(s) for computing the WAASBY index. Must be a numeric vector of the same length of resp. Defatul is 50, i.e., equal weights for stability and mean performance.
prob	The p-value for considering a IPCA significant.
naxis	The number of IPCAs to be used for computing the WAAS index. Default is NULL (Significant IPCAs are used). If values are informed, the number of IPCAS will be used independently on its significance. Note that if two or more variables are included in resp, then naxis must be a vector.
verbose	Logical argument. If verbose = FALSE the code are run silently.

Details

This function compute the weighted average of absolute scores, estimated as follows:

$$WAAS_i = \sum_{k=1}^{p} |IPCA_{ik} \times EP_k| / \sum_{k=1}^{p} EP_k$$

where $WAAS_i$ is the weighted average of absolute scores of the *i*th genotype; PCA_{ik} is the score of the *i*th genotype in the *k*th IPCA; and EP_k is the explained variance of the *k*th IPCA for $k = 1,2,..,p$, considering *p* the number of significant PCAs, or a declared number of PCAs. For example if prob = 0.05, all axis that are significant considering this probability level are used. The number of axis can be also informed by declaring naxis = x. This comand ignores the p.valuePC comand.

Value

individual	A within-environments ANOVA considering a fixed-effect model.
model	A data frame with the response variable, the scores of all Principal Components, the estimates of Weighted Average of Absolute Scores, and WAASY (the index that consider the weights for stability and productivity in the genotype ranking.
MeansGxE	The means of genotypes in the environments, with observed, predicted and residual values.
PCA	Principal Component Analysis.
anova	Joint analysis of variance for the main effects and Principal Component analysis of the interaction effect.
Details	A list summarizing the results. The following information are showed. WgtResponse, the weight for the response variable in estimating WAASB, WgtWAAS the weight for stability, Ngen the number of genotypes, Nenv the number of environments, OVmean the overall mean, Min the minimum observed (returning the genotype and environment), Max the maximum observed, Max the maximum observed, MinENV the environment with the lower mean, MaxENV the environment with the larger mean observed, MinGEN the genotype with the lower mean, MaxGEN the genotype with the larger.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

See Also

WAAS.AMMI

Examples

```
library(METAAB)

# Considering p-value <= 0.05 to compute the WAAS

model <- WAAS.AMMI(data_ge,
                   resp = GY,
                   gen = GEN,
                   env = ENV,
                   rep = REP)


# Declaring the number of axis to be used for computing WAAS
# and assigning a larger weight for the response variable when
# computing the WAASBY index.

model2 <- WAAS.AMMI(data_ge,
                    resp = GY,
                    gen = GEN,
                    env = ENV,
                    rep = REP,
                    naxis = 3,
                    wresp = 60)

# Analyzing multiple variables (GY and HM) at the same time
# considering that smaller values of HM are better and higher
# values of GY are better, assigning a larger weight for the GY
# and a smaller weight for HM when computing WAASBY index.

model3 <- WAAS.AMMI(data_ge,
                    resp = c(GY, HM),
                    gen = GEN,
                    env = ENV,
                    rep = REP,
                    mresp = c(100, 0),
                    wresp = c(60, 40))
```

WAASB   *Weighted Average of Absolute Scores for the BLUP'S GxE effects ma-*
                                        trix generated by a mixed-effect model

---

Description

Compute the Weighted Average of Absolute Scores for quantifying the stability in multienvironment trials using mixed-effect models.

Usage

```
WAASB(data, resp, gen, env, rep, mresp = NULL, wresp = NULL,
      random = "gen", prob = 0.05,  verbose = TRUE)
```

Arguments

data	The dataset containing the columns related to Environments, Genotypes, replication/block and response variable(s).
resp	The response variable(s). To analyze multiple variables in a single procedure a vector of variables may be used. For example resp = c(var1, var2, var3).
gen	The name of the column that contains the levels of the genotypes.
env	The name of the column that contains the levels of the environments.
rep	The name of the column that contains the levels of the replications/blocks.
mresp	A numeric vector of the same length of resp. The mresp will be the new maximum value after reescaling. By default, all variables in resp are rescaled so that de maximum value is 100 and the minimum value is 0.
wresp	The weight for the response variable(s) for computing the WAASBY index. Must be a numeric vector of the same length of resp. Defatul is 50, i.e., equal weights for stability and mean performance.
random	The effects of the model assumed to be random. Default is random = "gen" (genotype and genotype-vs-environment as random effects. Other values allowed are random = "env" (environment, genotype-vs-environment and block-within-environment random effects) or random = "all" all effects except the intercept are assumed to be random effects.
prob	The probability for estimating confidence interval for BLUP's prediction.
verbose	Logical argument. If verbose = FALSE the code are run silently.

Details

This function compute the weighted average of absolute scores considering all principal component axis from the Singular Value Decomposition (SVD) of the BLUP'S GxE effects matrix generated by a linear mixed-effect model. The main advantage of this procedure in relation to the WAAS.AMMI function is that random effects can be included in the model. In addition, unbalanced datasets can also be modeled.

Value

The function returns the results in a list for each analyzed variable. For each variable, the following objects are returned.

individual	A within-environments ANOVA considering a fixed-effect model.
fixed	Test for fixed effects.
random	Variance components for random effects.
LRT	The Likelihood Ratio Test for the random effects.
model	A data frame with the response variable, the scores of all Principal Components, the estimates of Weighted Average of Absolute Scores, and WAASY (the index that consider the weights for stability and productivity in the genotype ranking.
blupGEN	The estimated BLUPS for genotypes (If random = "gen" or random = "all")
BLUPenv	The estimated BLUPS for environments, (If random = "env" or random = "all").
BLUPge	The estimated BLUPS of all genotypes in all environments "BLUPij".
PCA	The results of Principal Component Analysis with eigenvalues and explained variance of BLUP-interaction matrix.

MeansGxE	The phenotypic means of genotypes in the environments, with observed, predicted and OLS residual prediction.
Details	A list summarizing the results. The following information are showed. WgtResponse, the weight for the response variable in estimating WAASB, WgtWAAS the weight for stability, Ngen the number of genotypes, Nenv the number of environments, OVmean the overall mean, Min the minimum observed (returning the genotype and environment), Max the maximum observed, Max the maximum observed, MinENV the environment with the lower mean, MaxENV the environment with the larger mean observed, MinGEN the genotype with the lower mean, MaxGEN the genotype with the larger.
ESTIMATES	A list with the following values: GEV the genotype-by-environment variance (and percentage of phenotypic variance); GV the genotypic variance (and percentage of phenotypic variance); EV the environmental variance;RV the residual variance (and percentage of phenotypic variance); FV the phenotypic variance; h2g the heritability of the trait; GEr2 the coefficient of determination of the interaction effects; h2mg the heritability of the mean; AccuGen the selective accuracy; rge the genotype-environment correlation; CVg the genotypic coefficient of variation; CVr the residual coefficient of variation; CVratio the ratio between genotypic and residual coefficient of variation.
residuals	The residuals of the model.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

See Also

WAAS.AMMI

Examples

```
library(METAAB)

# Genotypes as random effects and equal weights for both

# response variable and stability

model <-  WAASB(data_ge,
                resp = GY,
                gen = GEN,
                env = ENV,
                rep = REP,
                wresp = 70)

# Higher weight for response variable

model2 <- WAASB(data_ge,
                resp = GY,
                gen = GEN,
                env = ENV,
                rep = REP,
                wresp = 65)

# Environment as random effects analyzing more than one variables
```

```
# considering that smaller values of HM are better and higher
# values of GY are better, assigning a larger weight for the GY
# and a smaller weight for HM when computing WAASBY index.

model3 <- WAASB(data_ge,
                random = "env",
                resp = c(GY, HM),
                gen = GEN,
                env = ENV,
                rep = REP,
                mresp = c(100, 0),
                wresp = c(60, 40))
```

WAASBYratio	*Weighted Average of Absolute Scores from SVD of BLUP-interaction effects matrix in different scenarios of WAASB/GY ratio*

Description

Compute the Weighted Average of Absolute Scores for AMMI analysis in different combinations of weights for stability and productivity.

Usage

```
WAASBYratio(data, resp, gen, env, rep, increment = 5,
            saveWAASY = 50, progbar = TRUE)
```

Arguments

data	The dataset containing the columns related to Environments, Genotypes, replication/block and response variable(s).
resp	The response variable, for example `resp = RG`.
gen	The name of the column that contains the levels of the genotypes.
env	The name of the column that contains the levels of the environments.
rep	The name of the column that contains the levels of the replications/blocks.
increment	The range of the increment for WAASB/GY ratio. Default is 5. The function compute the WAASBY values starting with a weight o 100 for stability and 0 for response variable. With the default, the first scenario will be a WAASB/GY ratio = 100/0. In the next scenario, the WAASBY values are computed based on a WAASB/GY ratio = 95/5.
saveWAASY	Automatically save the WAASY values when the wheight for WAAS (stability) in the WAAS/GY ratio is "saveWAASY". Default is 100. The value of "saveWAASY" must be multiple of "Increment". If this assumption is not valid, an error will be occour.
progbar	A logical argument to define if a progress bar is shown. Default is `TRUE`.

Details

This function considers both stability (weighted average of absolute scores based on SVD of BLUP-interaction effects matrix) and productivitye for genotype ranking. This function provide the option of attributing weights for stability and productive in genotype ranking. This is important depending on the goal of a selection strategy. For example, if a a goal of a breeding program is to select a genotype whith high yielding (independeltly on the stability performance), that genotype with the first rank in an WAASB/GY = 0/100 ratio should be selected. The reciprocal is true. Aiming at selecting a high-stable genotype (independentely on the productivity), that genotype with the first rank in an WAASB/GY = 100/0 ratio should be selected. By defalut, the increment on the WAASB/GY ratio is equal to 5. In other words, twenty one different combinations are computed. Each combination, the genotypes are ranked regarding the WAASY value.

Value

anova	Joint analysis of variance for the main effects and Principal Component analysis of the interaction effect.
PC	Principal Component Analysis.
MeansGxE	The means of genotypes in the environments, with observed, predicted and residual values.
WAAS	A data frame with the response variable, the scores of all Principal Components, the estimates of Weighted Average of Absolute Scores, and WAASY (the index that consider the weights for stability and productivity in the genotype ranking.
WAASY	The values of the WAASY estimated when the wheight for the stability in the loop match with argument "saveWAASY".
WAASY.values	All the values of WAASY estimated in the different scenarios of WAAS/GY weighting ratio.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
library(METAAB)

# Default, with increment of 5 and saving the WAASY values when weight is 50
# wratio = WAASBYratio(data_ge,
#                      resp = GY,
#                      gen = GEN,
#                      env = ENV,
#                      rep = REP)

# Incrementing 2-by-2
wratio2 = WAASBYratio(data_ge,
                      resp = GY,
                      gen = GEN,
                      env = ENV,
                      rep = REP,
                      increment = 50)
```

WAASratio.AMMI	*Weighted Average of Absolute Scores for AMMI analysis in different scenarios of WAAS/GY ratio*

Description

Compute the Weighted Average of Absolute Scores for AMMI analysis in different combinations of weights for stability and productivity.

Usage

```
WAASratio.AMMI(data, resp, gen, env, rep, p.valuePC = 0.05,
               increment = 5, saveWAASY = 50, progbar = TRUE)
```

Arguments

data	The dataset containing the columns related to Environments, Genotypes, replication/block and response variable(s).
resp	The response variable, for example resp = RG.
gen	The name of the column that contains the levels of the genotypes.
env	The name of the column that contains the levels of the environments.
rep	The name of the column that contains the levels of the replications/blocks.
p.valuePC	The p-value for considering the PC significant. Default is 0.05. The number of significant Principal Components to be used for calculating the WAASB will be chosen based on this probability.
increment	The range of the increment for WAAS/GY ratio. Default is 5. The function compute the WAASY values starting with a weight o 100 for stability and 0 for response variable. With the default, the first scenario will be a WAAS/GY ratio = 100/0. In the next scenario, the WAASY values are computed based on a WAAS/GY ratio = 95/5.
saveWAASY	Automatically save the WAASY values when the wheight for WAAS (stability) in the WAAS/GY ratio is "saveWAASY". Default is 100. The value of "saveWAASY" must be multiple of "Increment". If this assumption is not valid, an error will be occour.
progbar	A logical argument to define if a progress bar is shown. Default is TRUE.

Details

This function is very similar to the WAASBYratio. The main difference is that here, the WAASBY values are computed considering a traditional AMMI model.

Value

anova	Joint analysis of variance for the main effects and Principal Component analysis of the interaction effect.
PC	Principal Component Analysis.
MeansGxE	The means of genotypes in the environments, with observed, predicted and residual values.

WAAS	A data frame with the response variable, the scores of all Principal Components, the estimates of Weighted Average of Absolute Scores, and WAASY (the index that consider the weights for stability and productivity in the genotype ranking.
WAASY	The values of the WAASY estimated when the wheight for the stability in the loop match with argument "saveWAASY".
WAASY.values	All the values of WAASY estimated in the different scenarios of WAAS/GY weighting ratio.

Author(s)

Tiago Olivoto <tiagoolivoto@gmail.com>

Examples

```
# Default, with increment of 5 and saving the WAASY values when weight is 50
#wratio = WAASratio.AMMI(data_ge,
#                        resp = GY,
#                        gen = GEN,
#                        env = ENV,
#                        rep = REP)

# Incrementing 2-by-2
wratio2 = WAASratio.AMMI(data_ge,
                         resp = GY,
                         gen = GEN,
                         env = ENV,
                         rep = REP,
                         increment = 50)
```

Index