

Package ‘gge’

May 15, 2018

Title Genotype Plus Genotype-by-Environment Biplots

Version 1.4

Date 2018-05-12

Description Create biplots for GGE (genotype plus genotype-by-environment) and GGB (genotype plus genotype-by-block-of-environments) models.

Type Package

Imports nipals, rgl, reshape2

Suggests agridat, knitr, lattice, testthat

License GPL-3

LazyData yes

URL <https://github.com/kwstat/gge>

BugReports <https://github.com/kwstat/gge/issues>

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Kevin Wright [aut, cre] (<<https://orcid.org/0000-0002-0617-8673>>),
Jean-Louis Laffont [aut]

Maintainer Kevin Wright <kw.stat@gmail.com>

Repository CRAN

Date/Publication 2018-05-15 07:28:18 UTC

R topics documented:

gge	2
RedGrayBlue	5
Index	7

 gge

GGE and GGB biplots

Description

GGE and GGB biplots

Fit a GGE (genotype + genotype * environment) model and display the results.

Usage

```
gge(x, ...)

## S3 method for class 'formula'
gge(formula, data = NULL, gen.group = NULL,
     env.group = NULL, ...)

## S3 method for class 'matrix'
gge(x, center = TRUE, scale = TRUE, gen.group = NULL,
     env.group = NULL, comps = c(1, 2), method = "svd", ...)

## S3 method for class 'gge'
plot(x, main = substitute(x), ...)

## S3 method for class 'gge'
biplot(x, main = substitute(x), subtitle = "",
        xlab = "auto", ylab = "auto", cex.gen = 0.6, cex.env = 0.5,
        col.gen = "darkgreen", col.env = "orange3", pch.gen = 1,
        lab.env = TRUE, comps = 1:2, flip = "auto", origin = "auto",
        res.vec = TRUE, hull = FALSE, zoom.gen = 1, zoom.env = 1, ...)

biplot3d(x, ...)

## S3 method for class 'gge'
biplot3d(x, cex.gen = 0.6, cex.env = 0.5,
         col.gen = "darkgreen", col.env = "orange3", comps = 1:3,
         lab.env = TRUE, res.vec = TRUE, zoom.gen = 1, ...)
```

Arguments

x	A matrix or data.frame.
...	Other arguments (e.g. maxiter, gramschmidt)
formula	A formula
data	Data frame
gen.group	genotype group
env.group	env group

center	If TRUE, center values for each environment
scale	If TRUE, scale values for each environment
comps	Principal components to use for the biplot. Default c(1,2).
method	method used to find principal component directions
main	Title, by default the name of the data. Use NULL to suppress the title.
subtitle	Subtitle to put in front of options. Use NULL to suppress the subtitle.
xlab	Label along axis. Default "auto" shows percent of variation explained. Use NULL to suppress.
ylab	Label along axis. Default "auto" shows percent of variation explained. Use NULL to suppress.
cex.gen	Character expansion for genotypes, default 0.6. Use 0 to omit genotypes.
cex.env	Character expansion for environments
col.gen	Color for genotypes
col.env	Color for environments
pch.gen	Plot character for genotypes
lab.env	Label environments if TRUE.
flip	If "auto" then each axis is flipped so that the genotype ordinate is positively correlated with genotype means. Can also be a vector like c(TRUE,FALSE) for manual control.
origin	If "auto", the plotting window is centered on genotypes, otherwise the origin is at the middle of the window.
res.vec	If TRUE, for each group, draw residual vectors from the mean of the locs to the individual locs.
hull	If TRUE, show a which-won-where polygon.
zoom.gen	Zoom factor for manual control of genotype xlim,ylim The default is 1. Values less than 1 may be useful if genotype names are long.
zoom.env	Zoom factor for manual control of environment xlim,ylim. The default is 1. Values less than 1 may be useful if environment names are long. Not used for 3D biplots.

Details

If there is replication in G*E, then the replications are averaged together before constructing the biplot.

The singular value decomposition of x is used to calculate the principal components for the biplot. Missing values are NOT allowed.

The argument `method` can be either 'svd' for complete-data or 'nipals' for missing-data.

Value

A list of class gge containing:

x	The filled-in data
x.orig	The original data
genCoord	genotype coordinates
locCoord	loc coordinates
blockCoord	block coordinates
gen.group	If not NULL, this specifies a classification of genotypes into groups.
env.group	If not NULL, this specifies a classification of environments into groups.
genMeans	genotype means
mosdat	mosaic plot data
R2	variation explained by each PC
center	Data centered?
scale	Data scaled?
method	Method used to calculate principal components.
pctMiss	Percent of x that is missing values
maxPCs	Maximum number of PCs

Author(s)

Kevin Wright, Jean-Louis Laffont
Jean-Louis Laffont, Kevin Wright

References

- Jean-Louis Laffont, Kevin Wright and Mohamed Hanafi (2013). Genotype + Genotype x Block of Environments (GGB) Biplots. *Crop Science*, 53, 2332-2341. <https://doi.org/10.2135/cropsci2013.03.0178>.
- Kroonenberg, Pieter M. (1997). *Introduction to Biplots for GxE Tables*, Research Report 51, Centre for Statistics, The University of Queensland, Brisbane, Australia. <http://three-mode.leidenuniv.nl/document/biplot.pdf>
- Yan, W. and Kang, M.S. (2003). *GGE Biplot Analysis*. CRC Press.

Examples

```
# Example 1. Data is a data.frame in 'matrix' format
B <- matrix(c(50, 67, 90, 98, 120,
              55, 71, 93, 102, 129,
              65, 76, 95, 105, 134,
              50, 80, 102, 130, 138,
              60, 82, 97, 135, 151,
              65, 89, 106, 137, 153,
              75, 95, 117, 133, 155), ncol=5, byrow=TRUE)
```

```
rownames(B) <- c("G1", "G2", "G3", "G4", "G5", "G6", "G7")
colnames(B) <- c("E1", "E2", "E3", "E4", "E5")

m1 = gge(B)
plot(m1)
biplot(m1, main="Example biplot")
# biplot3d(m1)

if(require(agridat)){
  # crossa.wheat biplot

  # Specify env.group as column in data frame
  data(crossa.wheat)
  dat2 <- crossa.wheat
  dat2$eg <- ifelse(is.element(dat2$loc,
                              c("KN", "NB", "PA", "BJ", "IL", "TC", "JM", "PI", "AS", "ID", "SC", "SS",
                                "SJ", "MS", "MG", "MM")), "Grp1", "Grp2")
  m2 <- gge(yield~gen*loc, dat2, env.group=eg, scale=FALSE)
  plot(m2)
  biplot(m2, lab.env=TRUE, main="crossa.wheat")
  # biplot3d(m2)
}
```

RedGrayBlue

Function to create a Red-Gray-Blue palette

Description

A function to create a Red-Gray-Blue palette.

Usage

```
RedGrayBlue(n)
```

Arguments

n Number of colors to create

Details

Using gray instead of white allows missing values to appear as white (actually, transparent).

Value

A vector of n colors.

Author(s)

Kevin Wright

Examples

```
pie(rep(1,11), col=RedGrayBlue(11))  
title("RedGrayBlue(11)")
```

Index

`biplot.gge` (`gge`), [2](#)

`biplot3d` (`gge`), [2](#)

`gge`, [2](#)

`gge-package` (`gge`), [2](#)

`gge.formula` (`gge`), [2](#)

`gge.matrix` (`gge`), [2](#)

`package-gge` (`gge`), [2](#)

`plot.gge` (`gge`), [2](#)

`RedGrayBlue`, [5](#)