Drawing RDA triplots

Description

This function produces triplots from the output list of the RDA function 'rdaTest'.

Usage

graph.rdaTest(
        rdaTest.out, xax=1, yax=2, scaling=1, graph.type="notchosen", plot.sites=TRUE,
        plot.spc=TRUE, plot.env=TRUE, binary=NULL, height=6, width=6, xlim=NULL,
        ylim=NULL, ell=NULL, alpha=0.95, ell.axis=FALSE, mul.spc=0.8, mul.env=0.90,
        mul.text=0.10, pos.ell=NULL, pos.bin=NULL, pos.sites=1, label.sites=TRUE,
        label.spc=TRUE, label.env=TRUE, label.ell=TRUE, col.env="blue", col.spc="red",
        col.site="black", col.ell="black", lty.env=1, lty.spc=1, lty.ell=1, lty.axis=2, cex=1,
        cex.lab=1, cex.axis=1, lwd=1, len=0.1, saveplot=FALSE, path=NULL, mar.perc=NULL)

Arguments

rdaTest.out	name of the rdaTest output object. The most simple call to this function is to provide only the 'rdaTest.out' output object name. Ex.: graph.rdaTest(toto).
xax, yax	canonical axes that will be used for the triplot. These value can be negative; if so, the plot axes will be reversed.
scaling	allows the user to choose between the two scaling types, 1 or 2, for the triplots: scaling = 1: distance triplot (default); scaling = 2: correlation triplot.
graph.type	either "F" or "Z" -- "F"= site scores, "Z" = fitted site scores.
plot.sites	if TRUE, the sites will be plotted as small circles.
plot.spc	if TRUE, the species (or other response variables) will be plotted.
plot.env	if TRUE, the explanatory (environmental) variables will be plotted.
binary	a vector specifying which of the environmental variables are binary; the corresponding data points will be averaged and represented by a symbol instead of an arrow, as in Canoco.
saveplot	• if set to FALSE, the graph will only appear in a window (quartz); • if set to TRUE, no graph will be plotted on the screen; a pdf file will created.
path	complete path to the file in which the plot will be saved; example: "~/Desktop/toto.pdf"
width, height	size of the triplot in inches.
xlim, ylim	vectors describing the minimum and maximum values of the plotted region.
lwd	line width for the axes, arrows, and ellipses.
len	length of the arrow heads.
mar.perc	allows to increase or decrease the margin size, using a percentage of the total width and height of the graph; the value must be between 0 and 1.

Parameters controlling the confidence ellipses around groups of points

ell	draw confidence ellipses around groups of points according to a vector defining the group assignments of the objects; see section **Examples** below.
alpha	confidence region of the ellipses (e.g. 0.90, 0.80, etc.) Default value: 0.95.
lty.ell	drawing type for ellipses; see 'lty' in the help documentation file of 'par'.
col.ell	the ellipses and group identification text color; see 'lty' in the help file of 'par'.
ell.axis	if TRUE, draw the major and minor axes of the ellipses.

| pos.ell | offset the group names relative to the ellipse centers: 1=bottom, 2=left, 3=top, 4=right. |

Parameters for arrows and text

label.sites	if TRUE, labels are added to the site symbols.
label.spc	if TRUE, labels are added to the species arrows.
label.env	if TRUE, labels are added to the environmental variable arrows.
label.ell	if TRUE, labels (numbers) are added to the group ellipses.
col.env	color for the environmental variable arrows and text.
col.spc	color for the species arrows and text.
col.site	color for the site points and text.
lty.env	environmental variable arrows drawing type; see 'lty' in the help documentation file of 'par'.
lty.spc	species arrows drawing type; see 'lty' in the help documentation file of 'par'.
lty.axis	drawing type for axes, full lines, dotted lines, etc.
cex.xxx	change of the font size, with respect to a standard of 1.
mul.spc	multiplier based on the range of the plot axes, which changes to the length of the species arrows proportionally; 1 = maximum relative length in the set of arrows.
mul.env	same, for the explanatory variables
mul.text	text position at the end of each arrow plus a percentage; default is 10%.
pos.bin	offset the names of the binary variables: 1=bottom, 2=left, 3=top, 4=right.
pos.sites	offset the site labels: 1=bottom, 2=left, 3=top, 4=right.

Details

A triplot of the results of the canonical redundancy analysis (RDA) is drawn, using either scaling 1 or scaling 2.

Scaling type 1 triplot uses matrices Z or F for objects and U for variables; notation as in Legendre and Legendre (1998, Section 11.1.1). This projection preserves the Euclidean distances among the objects.

Scaling type 2 triplot uses matrices ZSc2 or FSc2 for objects and USc2 for variables. This projection preserves the correlations among the response variables' fitted values.

Value

The function plots a canonical ordination triplot to a graphics window or a PDF file.

Reference

Legendre, P. and Legendre, L. 1998. *Numerical Ecolog*y. 2nd English ed. Elsevier, Amsterdam.

Author(s)

Sébastien Durand and Pierre Legendre, Département de sciences biologiques, Université de Montréal.

See Also

rdaTest

Examples

```
## Example from Legendre and Legendre (1998), p. 590, Table 11.3
Table11.3 = read.table(file.choose())          # File "Table_11-3.txt"
Y = Table11.3[,1:6]
X = Table11.3[,7:10]
result = rdaTest(Y, X, testF=TRUE, nperm=999)
graph.rdaTest(result, graph.type="Z")

## Reverse the orientation of the abscissa ('xax') in the plot
graph.rdaTest(result, xax=-1, yax=2, mul.spc=0.90, mul.env=0.70, mul.text=0.10, scaling=1,
graph.type="F", mai.perc=0.15, pos.sites=3)

## Draw confidence ellipses around groups of points, spider data
spiders.spe = read.table("Spiders_28x12_spe.txt")
spiders.hel = decostand(spiders.spe, "hellinger")
spiders.env = read.table("Spiders_28x4_env.txt")
result.spiders = rdaTest(spiders.hel, spiders.env)
# 'vec' divides the sites into three a priori group
vec = c(2,1,2,1,1,3,1,3,2,2,2,2,1,3,3,3,3,3,3,3,2,2,2,3,2,2,2)
graph.rdaTest(result.spiders, graph.type="Z", ell=vec, lty.ell=3, pos.sites=4, mar.perc=0.15)

## Plot the sites and environmental variables, but not the species
graph.rdaTest(result.spiders, graph.type="Z", plot.spc=FALSE, ell=vec, lty.ell=3, pos.sites=4,
mar.perc=0.15)
```

```
## Draw confidence ellipses around groups of points, mite data
library(vegan)
data(mite)
data(mite.env)
mite.hel = decostand(mite, 'hellinger')
mm = model.matrix(~ SubsDens + WatrCont + Substrate + Shrub + Topo, mite.env)[,-1]
result.mite = rdaTest(mite.hel, mm)
# 'vec' divides the sites in 4 groups according to the values of water content (variable WatrCont)
vec= c(2,4,2,2,1,2,2,1,2,1,1,2,1,1,2,2,2,2,2,1,1,1,2,2,2,4,1,2,2,2,4,2,2,2,2,4,2,4,3,2,4,2,3,3,4,
4,2,4,2,4,4,3,4,4,4,4,3,2,3,4,3,3,4,4,4,4,3,3,4,4)
graph.rdaTest(result.mite, graph.type="Z", ell=vec, lty.ell=3, pos.sites=4, mar.perc=0.15)
### Do not draw the species
graph.rdaTest(result.mite, graph.type="Z", ell=vec, lty.ell=3, pos.sites=4, mar.perc=0.15,
col.spc="white")
```