

Package ‘bivariatemaps’

July 22, 2025

Title Creates Bivariate Maps

Version 1.7

Description Contains functions mainly focused to plotting bivariate maps.

License GPL-3

Encoding UTF-8

Imports classInt, terra, base, graphics, stats, caper, picante, ape,
betapart, CommEcol, grDevices

RoxygenNote 7.2.3

NeedsCompilation no

Author Jose Hidasi-Neto [aut, cre]

Maintainer Jose Hidasi-Neto <hidasineto@gmail.com>

Repository CRAN

Date/Publication 2024-10-04 22:10:02 UTC

Contents

ancestree	2
astRaRium	2
betagrid	3
bivariate.map	4
colmat	5
EcoEDGE	6
GridFilter	7
tempbetagrid	7
Index	9

ancestree	<i>ancestree: Get a Phylogeny that Starts at the Common Ancestor of a Group of Species</i>
-----------	--

Description

Uses a sample of species to get the phylogeny that starts at the common ancestor of the sample.

Usage

```
ancestree(sample, tree)
```

Arguments

sample	A data frame with species in columns' names (check out example link).
tree	.tre phylogenetic tree of species.

Value

A phylogeny that starts at the common ancestor of the provided sample.

Examples

```
# https://rfunctions.blogspot.com/2013/04/ancestree-function-get-phylogeny-that.html
```

astRaRium	<i>astRaRium: Generate infinite Astrarium minigames.</i>
-----------	--

Description

Generate infinite minigames based on the Astrarium minigame, from "Dragon Age: Inquisition". You have to click on the stars (asterisks) to find a path that that reveals the same image as shown while not repeating connections already made once. When it begins (by typing "astRaRium()"), it will ask you for the desired number of stars and the number of links between pairs of stars. A good number is 12 and 9, respectively.

Usage

```
astRaRium()
```

Value

You play the game. You can succeed or fail.

Examples

```
# https://rfunctions.blogspot.com/2021/05/astrium2-r-game-about-linking-stars.html
```

betagrid	<i>betagrid: Calculates Beta Diversity for all the Grid Cells in Comparison with Their Nearest Neighbor Cells</i>
----------	---

Description

Uses a 'gridded' shapefile and its corresponding community data matrix to calculate beta diversity for each focal grid cell in relation to its neighbor cells (for example, 8 nearest neighbors).

Usage

```
betagrid(
  gridshp,
  comp,
  xfeature,
  yfeature,
  radius,
  phylotree,
  phylobeta = F,
  index = "sorensen"
)
```

Arguments

gridshp	Gridded Shapefile ("SpatialPolygonsDataFrame").
comp	community data matrix (species occurrence on each grid cell).
xfeature	number of the feature within the gridded shapefile corresponding to the longitude.
yfeature	number of the feature within the grid shapefile corresponding to the latitude.
radius	the radius (in degrees) that define the maximum distance to select neighbor cells (for example, use the resolution of the gridded shapefile).
phylotree	optional phylogenetic tree ("phylo" class). It can also be a "phylo" class functional dendrogram.
phylobeta	Optional boolean whether or not to calculate or not phylogenetic beta diversity (see "phylo.beta.pair" function in "betapart" package) instead of the usual beta diversity (see "beta.pair" function in "betapart" package). Default is F.
index	to calculate "sorensen" or "jaccard". Default is "sorensen".

Value

A data frame with values of mean turnover, mean nestedness, and mean beta diversity for each grid cell.

Examples

```
# https://rfunctions.blogspot.com/2015/08/calculating-beta-diversity-on-grid.html
```

bivariate.map	<i>bivariate.map: Create a Bivariate Map</i>
---------------	--

Description

Creates a Bivariate Map using two rasters and a color matrix created with colmat() function.

Usage

```
bivariate.map(  
  rasterx,  
  rastery,  
  colormatrix,  
  nquantiles = 10,  
  trivariate = F,  
  rasterz  
)
```

Arguments

rasterx	raster
rastery	raster
colormatrix	color matrix from colmat() function
nquantiles	number of quantiles in color matrix (same as used when using colmat() function)
trivariate	boolean indicating if should use a third raster for a "trivariate" map where the third variable corresponds to the transparency (alpha) of colors.
rasterz	raster (only if trivariate=T)

Value

A plot with the bivariate map.

Examples

```
# https://rfunctions.blogspot.com/2015/03/bivariate-maps-bivariatemap-function.html
```

colmat*colmat: Create a Color Matrix*

Description

Creates a color matrix to be used in `bivariate.map()` function.

Usage

```
colmat(  
  nquantiles = 10,  
  upperleft = "blue",  
  upperright = "red",  
  bottomleft = "grey",  
  bottomright = "yellow",  
  xlab = "x label",  
  ylab = "y label"  
)
```

Arguments

<code>nquantiles</code>	numeric variable for number of quantiles in color matrix
<code>upperleft</code>	upperleft color of color matrix
<code>upperright</code>	upperright color of color matrix
<code>bottomleft</code>	bottomleft color of color matrix
<code>bottomright</code>	bottomright color of color matrix
<code>xlab</code>	character variable
<code>ylab</code>	character variable

Value

Two outputs: a color matrix object to be used in `bivariate.map()` function, and a plot of the color matrix.

Examples

```
col.matrix<-colmat(nquantiles=10, xlab="My x label", ylab="My y label")  
  
# https://rfunctions.blogspot.com/2015/03/bivariate-maps-bivariatemap-function.html
```

EcoEDGE

*EcoEDGE: Ecologically and Evolutionarily Distinct and Globally Endangered***Description**

Calculates ED, EcoD, EDGE, EcoDGE, and EcoEDGE, from Red List Categories, a phylogenetic tree, and a functional dendrogram.

Usage

```
EcoEDGE(
  category,
  phylo,
  func,
  funcweight = 0.5,
  phyloweight = 0.5,
  polytomy = "isaac"
)
```

Arguments

category	data frame where the first column contains the red list categories of species.
phylo	phylogenetic tree of species.
func	functional dendrogram of species.
funcweight	weight of ecological distinctiveness on the calculation of EcoEDGE scores.
phyloweight	weight of evolutionary distinctiveness on the calculation of EcoEDGE scores. Sum of funcweight and phyloweight must be equal to 1.
polytomy	how to handle polytomies in the phylogeny and the functional dendrogram. See "ed.calc" function for details.

Value

A table with calculated values for each variable.

Examples

```
# https://rfunctions.blogspot.com/2021/03/ecoedge-ecologically-and-evolutionarily.html
```

GridFilter	<i>GridFilter: Intersect a Shape with a Grid and Exclude Cells Based on Area Coverage</i>
------------	---

Description

Overlay a shapefile with a grid. Then, excludes grid cells that do not cover a specific amount of the Shape area.

Usage

```
GridFilter(shape, resol = 1, prop = 0)
```

Arguments

shape	A shapefile.
resol	The resolution (in degrees) of the grid you want to overlay. The default is 1 degree of longitude and latitude.
prop	The minimum value of area covered by the grid cell. The default is 0 (i.e. it does not delete any grid cell).

Value

A gridded shapefile.

Examples

```
# https://rfunctions.blogspot.com/2014/12/gridfilter-intersect-grid-with-shape.html
```

tempbetagrid	<i>tempbetagrid: Calculates Temporal Beta Grid Using Species' Occurrences</i>
--------------	---

Description

Uses two occurrence matrices (from different time periods) to calculate temporal beta diversity for each site or grid cell.

Usage

```
tempbetagrid(oc1, oc2, index = "sorensen", phylotree, phylobeta = F)
```

Arguments

oc1	Community data matrix (species occurrence on each grid cell).
oc2	Second community data matrix (species occurrence on each grid cell) for another time period.
index	Calculate "sorensen" or "jaccard". Default is "sorensen".
phylotree	Optional phylogenetic tree ("phylo" class). It can also be a functional dendrogram.
phylobeta	Optional boolean indicating whether or not to calculate phylogenetic beta diversity (see "phylo.beta.pair" function in "betapart" package) instead of the usual beta diversity (see "beta.pair" function in "betapart" package). Default is F.

Value

A data frame with values of turnover, nestedness, beta diversity, and turnover/beta for each site or grid cell.

Examples

```
# https://rfunctions.blogspot.com/2016/10/calculating-temporal-beta-diversity-on.html
```


Index

ancestree, [2](#)

astRaRium, [2](#)

betagrid, [3](#)

bivariate.map, [4](#)

colmat, [5](#)

EcoEDGE, [6](#)

GridFilter, [7](#)

tempbetagrid, [7](#)