Package: phylosignal (via r-universe)

October 6, 2024

Type Package

Title Exploring the Phylogenetic Signal in Continuous Traits

Version 1.3.1

Date 2023-10-11

Author Francois Keck <francois.keck@gmail.com>

Maintainer Francois Keck <francois.keck@gmail.com>

Description A collection of tools to explore the phylogenetic signal in univariate and multivariate data. The package provides functions to plot traits data against a phylogenetic tree, different measures and tests for the phylogenetic signal, methods to describe where the signal is located and a phylogenetic clustering method.

License GPL-3

Imports Rcpp (>= 0.11.0), adephylo, igraph, ape, phylobase, boot, DBI, graphics, grDevices, methods, stats, utils

LinkingTo Rcpp, RcppArmadillo

Suggests testthat, knitr, rmarkdown

RoxygenNote 7.2.1

LazyLoad yes

LazyData true

Encoding UTF-8

VignetteBuilder knitr

Repository https://phylotastic.r-universe.dev

RemoteUrl https://github.com/fkeck/phylosignal

RemoteRef HEAD

RemoteSha dfa59275a58a9e6e941e277b3499e084c7b0575c

Contents

`.layouterize` *Set layout for plots*

Description

Set layout for plots

Usage

```
.layouterize(n.traits, show.tip)
```

Arguments

`n.traits`	the number of traits in the layout
`show.tip`	a logical indicating whether tip names are included in the layout

`.layouterizeRatio` *Set layout for plots*

Description

Set layout for plots

Usage

```
.layouterizeRatio(tree.ratio, n.traits, show.tip)
```

Arguments

`tree.ratio`	the ratio of phylogenetic tree included in the layout
`n.traits`	the number of traits in the layout
`show.tip`	a logical indicating whether tip names are included in the layout

`.orderGrArg` *Reordering vector or matrix of settings.*

Description

Internal function of multiplot.phylo4d. Reordering vector or matrix of settings and check for names consistencies.

Usage

```
.orderGrArg(x, n.tips, n.traits, new.order, tips, default)
```

Arguments

`x`	a vector or a matrix to order.
`n.tips`	number of tips.
`n.traits`	number of traits
`new.order`	a numeric vector giving the new order.
`tips`	a character vector giving the tips labels (with the new order)
`default`	the default value

Value

An ordered vector or matrix. An error if problem of consistency.

`barplot.phylo4d` *Barplot of Traits Values along a Phylogeny*

Description

Barplot of Traits Values along a Phylogeny

Usage

```
## S3 method for class 'phylo4d'
barplot(
  height,
  trait = names(tdata(height)),
  center = TRUE,
  scale = TRUE,
  tree.ladderize = FALSE,
  tree.type = "phylogram",
  tree.ratio = NULL,
  tree.xlim = NULL,
  tree.open.angle = 0,
  tree.open.crown = TRUE,
  show.tip = TRUE,
  tip.labels = NULL,
  tip.col = "black",
  tip.cex = 1,
  tip.font = 3,
  tip.adj = 0,
  data.xlim = NULL,
  bar.lwd = 10,
  bar.col = "grey35",
  show.data.axis = TRUE,
  show.trait = TRUE,
  trait.labels = NULL,
  trait.col = "black",
  trait.cex = 1,
  trait.font = 1,
  trait.bg.col = "grey90",
  error.bar.sup = NULL,
  error.bar.inf = NULL,
  error.bar.col = 1,
  show.box = FALSE,
  grid.vertical = TRUE,
  grid.horizontal = FALSE,
  grid.col = "grey25",
  grid.lty = "dashed",
  ...
)
```

Arguments

`height` a `phylo4d` object.

`trait` the traits in the `phylo4d` object to include in the plot. Can be a character vector giving the name of the traits or numbers giving the column index in the table of the data slot of the p4d object. Can be used to reorder the traits in the plot.

`center` a logical indicating whether traits values should be centered.

`scale` a logical indicating whether traits values should be scaled.

`tree.ladderize` a logical indicating whether the tree should be (right) ladderized.

`tree.type` a character string specifying the type of phylogeny to be drawn. Can be "`phylogram`", "`cladogram`" or "`fan`".

`tree.ratio` a numeric value in [0, 1] giving the proportion of width of the figure for the tree.

`tree.xlim` a numeric vector of length 2 giving the limits of the x-axis for the tree. If `NULL`, it is determined automatically.

`tree.open.angle`

a numeric value giving the angle in degrees left blank if `tree.type = "fan"`.

`tree.open.crown`
a logical indicating whether the crowns should be drawn following the value of `tree.open.angle` (default `TRUE`).

`show.tip` logical indicating whether tips labels should be drawn.

`tip.labels` character vector to label the tips. If `NULL` the tips labels of the `phylo4d` object are used

`tip.col` a vector of R colors to use for the tips labels. Recycled if necessary.

`tip.cex` a numeric vector to control character size of the tips labels. Recycled if necessary.

`tip.font` an integer vector specifying the type of font for the tips labels: 1 (plain text), 2 (bold), 3 (italic), or 4 (bold italic). Recycled if necessary.

`tip.adj` a vector of numeric in [0, 1] to control tips labels justification: 0 (left-justification), 0.5 (centering), or 1 (right-justification). Recycled if necessary.

`data.xlim` numeric vector of length 2 or matrix giving the x coordinates range for the barplots/dotplots (see Details).

`bar.lwd` a vector of numeric giving bar widths of the barplot(s). Recycled along the tips, reapeated for each trait.

`bar.col` a vector of R colors to use for the bars. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning (see Details)

`show.data.axis` logical indicating whether barplots/dotplots axes should be drawn.

`show.trait` logical indicating whether traits labels should be drawn.

`trait.labels` character vector to label the traits. If `NULL` the traits labels of the `phylo4d` object are used.

`trait.col` a vector of R colors to use for the traits labels. Recycled if necessary.

`trait.cex` a numeric vector to control character size of the trait labels. Recycled if necessary.

`trait.font` an integer vector specifying the type of font for the traits labels: 1 (plain text), 2 (bold), 3 (italic), or 4 (bold italic). Recycled if necessary.

`trait.bg.col` a vector of R colors to use for the background of the barplots. Recycled if necessary.

`error.bar.sup` a matrix giving the superior limit for error bars. Columns and rows names must match with traits and tips labels, respectively.

`error.bar.inf` a matrix giving the inferior limit for error bars. Columns and rows names must match with traits and tips labels, respectively.

`error.bar.col` a vector of R colors to use for the bars. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning (see Details)

`show.box` a logical indicating whether a box should be drawn around the plots.

`grid.vertical` a logical incating whether vertical lines of the grid should be drawn.

`grid.horizontal`
a logical incating whether horizontal lines of the grid should be drawn.

`grid.col` a vector of R colors to use for the lines of the grid.

`grid.lty` the lines type of the grid. Possibly a vector.

`...` further arguments to be passed to `plot.phylo`.

Examples

```
data(navic)
barplot(navic)
```

`descentsNames` *Names of descents*

Description

This function return all the names of the descents (tips) of a given node.

Usage

```
descentsNames(phy, node)
```

Arguments

`phy`	an object of class "`phylo`.
`node`	the ID of the node.

Value

A vector tip labels.

`directDescents` *ID of direct descents*

Description

This function return the node number (ID) of the direct descents of a given node

Usage

```
directDescents(phy, node)
```

Arguments

`phy`	an object of class "`phylo`.
`node`	the ID of the node.

Value

A vector of node numbers.

`distEq` *Pairwise Distance from Regularly Distributed Points*

Description

This function return a distance (or proximity) matrix between n points regularly distributed in 1 dimension.

Usage

```
distEq(n, prox = FALSE)
```

Arguments

`n`	the number of points
`prox`	a logical indicating whether to return a matrix of proximity. Default to `FALSE` so the function returns the matrix of distance.

Value

A squared matrix. A matrix of class `dist` if `prox` is set to `FALSE`.

Examples

```
x <- distEq(5)
```

`dotplot` *Dotplot*

Description

Dotplot

Usage

```
dotplot(...)
```

Arguments

`...`	further arguments passed to or from other methods.

dotplot.phylo4d *Dotplot of Traits Values along a Phylogeny*

Description

Dotplot of Traits Values along a Phylogeny

Usage

```
## S3 method for class 'phylo4d'
dotplot(
  p4d,
  trait = names(tdata(p4d)),
  center = TRUE,
  scale = TRUE,
  tree.ladderize = FALSE,
  tree.type = "phylogram",
  tree.ratio = NULL,
  tree.xlim = NULL,
  tree.open.angle = 0,
  tree.open.crown = TRUE,
  show.tip = TRUE,
  tip.labels = NULL,
  tip.col = "black",
  tip.cex = 1,
  tip.font = 3,
  tip.adj = 0,
  data.xlim = NULL,
  show.data.axis = TRUE,
  dot.col = "black",
  dot.pch = 20,
  dot.cex = 2,
  show.trait = TRUE,
  trait.labels = NULL,
  trait.col = "black",
  trait.cex = 1,
  trait.font = 1,
  trait.bg.col = "grey90",
  error.bar.sup = NULL,
  error.bar.inf = NULL,
  error.bar.col = 1,
  show.box = FALSE,
  grid.vertical = FALSE,
  grid.horizontal = TRUE,
  grid.col = "grey25",
  grid.lty = "dashed",
  ...
)
```

Arguments

`p4d` a `phylo4d` object.

`trait` the traits in the `phylo4d` object to include in the plot. Can be a character vector giving the name of the traits or numbers giving the column index in the table of the data slot of the p4d object. Can be used to reorder the traits in the plot.

`center` a logical indicating whether traits values should be centered.

`scale` a logical indicating whether traits values should be scaled.

`tree.ladderize` a logical indicating whether the tree should be (right) ladderized.

`tree.type` a character string specifying the type of phylogeny to be drawn. Can be `"phylogram"`, `"cladogram"` or `"fan"`.

`tree.ratio` a numeric value in [0, 1] giving the proportion of width of the figure for the tree.

`tree.xlim` a numeric vector of length 2 giving the limits of the x-axis for the tree. If `NULL`, it is determined automatically.

`tree.open.angle`
a numeric value giving the angle in degrees left blank if `tree.type = "fan"`.

`tree.open.crown`
a logical indicating whether the crowns should be drawn following the value of `tree.open.angle` (default `TRUE`).

`show.tip` logical indicating whether tips labels should be drawn.

`tip.labels` character vector to label the tips. If `NULL` the tips labels of the `phylo4d` object are used

`tip.col` a vector of R colors to use for the tips labels. Recycled if necessary.

`tip.cex` a numeric vector to control character size of the tips labels. Recycled if necessary.

`tip.font` an integer vector specifying the type of font for the tips labels: 1 (plain text), 2 (bold), 3 (italic), or 4 (bold italic). Recycled if necessary.

`tip.adj` a vector of numeric in [0, 1] to control tips labels justification: 0 (left-justification), 0.5 (centering), or 1 (right-justification). Recycled if necessary.

`data.xlim` numeric vector of length 2 or matrix giving the x coordinates range for the barplots/dotplots (see Details).

`show.data.axis` logical indicating whether barplots/dotplots axes should be drawn.

`dot.col` a vector of R colors to use for the points. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning (see Details)

`dot.pch` a numerical vector of symbol to use for the points. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning (see Details)

`dot.cex` a numerical vector. Character (or symbol) expansion for the points. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning (see Details) `show.trait` logical indicating whether traits labels should be drawn.

`trait.labels` character vector to label the traits. If `NULL` the traits labels of the `phylo4d` object are used.

`trait.col` a vector of R colors to use for the traits labels. Recycled if necessary.

`trait.cex` a numeric vector to control character size of the trait labels. Recycled if necessary.

`trait.font` an integer vector specifying the type of font for the traits labels: 1 (plain text), 2 (bold), 3 (italic), or 4 (bold italic). Recycled if necessary.

`trait.bg.col` a vector of R colors to use for the background of the barplots. Recycled if necessary.

`error.bar.sup` a matrix giving the superior limit for error bars. Columns and rows names must match with traits and tips labels, respectively.

`error.bar.inf` a matrix giving the inferior limit for error bars. Columns and rows names must match with traits and tips labels, respectively.

`error.bar.col` a vector of R colors to use for the bars. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning (see Details)

`show.box` a logical indicating whether a box should be drawn around the plots.

`grid.vertical` a logical incating whether vertical lines of the grid should be drawn.

`grid.horizontal`
a logical incating whether horizontal lines of the grid should be drawn.

`grid.col` a vector of R colors to use for the lines of the grid.

`grid.lty` the lines type of the grid. Possibly a vector.

`...` further arguments to be passed to `plot.phylo`.

Examples

```
data(navic)
dotplot(navic)
```

`evenColors` *Palette of evenly distributed colors*

Description

This function generates a vector of n colors evenly distributed in the RGB space. Usefull to create palettes of distinct colors.

Usage

```
evenColors(n)
```

Arguments

`n`	the number of colors to be in the palette.

Value

a vector of hexadecimal colors.

`focusTraits` *Focus on sub parts of a plot*

Description

These functions can be used after `barplot.phylo4d`, `dotplot.phylo4d` and `gridplot.phylo4d` when `tree.type` is `"phylogram"` or `"cladogram"` to focus on the different part of the plot and add graphical elements.

Usage

```
focusTraits(x)

focusTree()

focusTips()

focusStop()
```

Arguments

`x`	the trait to focus on. Can be a character string giving the name of the trait or an integer giving the number of the trait in order of appearance in the plot.

Details

#' Use `focusTree` to focus on the phylogenetic tree, `focusTraits` to focus on a given trait and `focusTips` to focus on the tips labels. Use `focusStop` to close the editing and restore graphical settings. For each part of the plot, the coordinate system is restored, making edition easier. For the phylogeny, post-editing functions of the package `ape` like `nodelabels` can be used.

Examples

```
require(ape)
require(phylobase)
data(navic)
dat <- tdata(navic)
neidium.sp <- c("Neidium bisulcatum",
                "Neidium affine",
                "Neidium productum")
stauroneis.sp <- c("Stauroneis kriegeri",
                      "Stauroneis acuta",
                      "Stauroneis gracilior",
                      "Stauroneis phoenicenteron")
neidium.mean <- mean(dat[neidium.sp,])
stauroneis.mean <- mean(dat[stauroneis.sp, ])

dotplot(navic, center = FALSE, scale = FALSE, data.xlim= c(0, 6))

focusTree()
nodelabels(node=c(22, 32), pch = 20, cex = 3, col = c(2, 3))

focusTraits()
segments(x0 = neidium.mean, y0 = 14.5,
         x1 = neidium.mean, y1 = 17.5,
         col = 3, lty = "dashed", lwd = 2)
segments(x0 = stauroneis.mean, y0 = 2.5,
         x1 = stauroneis.mean, y1 = 7.5,
         col = 2, lty = "dashed", lwd = 2)

focusTips()
rect(xleft = 0, ybottom = 2.5,
     xright = 0.9, ytop = 7.5,
     col = "#FF000020", border = NA)
rect(xleft = 0, ybottom = 14.5,
     xright = 0.9, ytop = 17.5,
     col = "#00FF0020", border = NA)

focusStop()
```

graphClust *Phylogenetically constrained clustering*

Description

This function extracts clusters of species based on traits values and phylogenetic proximities.

Usage

```
graphClust(
  p4d,
  trait = names(tdata(p4d)),
  lim.phylo = 0.2,
  lim.trait = 0.2,
  select.method = "ellipse",
  dist.phylo = "patristic",
  dist.trait = "euclidean",
  scale.lim = TRUE
)
```

Arguments

`p4d` a `phylo4d` object.

`trait` the traits in the `phylo4d` object to use for clustering. Can be a character vector giving the name of the traits or numbers giving the column index in the table of the data slot of the `phylo4d` object.

`lim.phylo` the maximum phylogenetic distance for edges selection.

`lim.trait` the maximum trait-based distance for edges selection.

`select.method` a character string specifying the method used to select edges. This must be one of `"line"`, `"rectangle"` or `"ellipse"`.

`dist.phylo` a matrix of phylogenetic distances or a character string specifying a method to compute it. See Details.

`dist.trait` a character string specifying the method used to compute traits distances. See Details.

`scale.lim` logical (default `TRUE`) indicating if `lim.phylo` and `lim.trait` are scaled (divided by their max value).

Details

If `"dist.phylo"` is a character string, the phylogenetic distance matrix is computed internally using the function `distTips` from the package **adephylo**. All the methods supported by `distTips` are available: `"patristic"`,`"nNodes"`,`"Abouheif"` and `"sumDD"`. See `distTips` for details about the methods.

If `"dist.trait"` is a character string, the traits distance matrix is computed with the `dist` function. All the methods supported by `dist` are available: `"euclidean"`,`"maximum"`, `"manhattan"`, `"canberra"`, `"binary"` and `"minkowski"`. See `dist` for details about the methods.

Value

An object of class `graphclust`.

Examples

```
data(navic)
gC <- graphClust(navic, lim.phylo = 1, lim.trait = 2, scale.lim = FALSE)
gC
plot.graphclust(gC, which = "selection", ask = FALSE)
plot.graphclust(gC, which = "graph", ask = FALSE)
plot.graphclust(gC, which = "tree", ask = FALSE)
```

`gridplot` *Gridplot*

Description

Gridplot

Usage

```
gridplot(...)
```

Arguments

`...`	further arguments passed to or from other methods.

`gridplot.phylo4d` *Gridplot of Traits Values along a Phylogeny*

Description

Gridplot of Traits Values along a Phylogeny

Usage

```
## S3 method for class 'phylo4d'
gridplot(
  p4d,
  trait = names(tdata(p4d)),
  center = TRUE,
  scale = TRUE,
  tree.ladderize = FALSE,
  tree.type = "phylogram",
  tree.ratio = NULL,
  tree.xlim = NULL,
  tree.open.angle = 0,
  tree.open.crown = TRUE,
  show.tip = TRUE,
  tip.labels = NULL,
  tip.col = "black",
  tip.cex = 1,
  tip.font = 3,
  tip.adj = 0,
  cell.col = white2red(100),
  show.color.scale = TRUE,
  show.trait = TRUE,
  trait.labels = NULL,
  trait.col = "black",
  trait.cex = 0.7,
  trait.font = 1,
  trait.bg.col = "grey90",
  show.box = FALSE,
  grid.vertical = FALSE,
  grid.horizontal = FALSE,
  grid.col = "grey25",
  grid.lty = "dashed",
  ...
)
```

Arguments

`p4d` a `phylo4d` object.

`trait` the traits in the `phylo4d` object to include in the plot. Can be a character vector giving the name of the traits or numbers giving the column index in the table of the data slot of the p4d object. Can be used to reorder the traits in the plot.

`center` a logical indicating whether traits values should be centered.

`scale` a logical indicating whether traits values should be scaled.

`tree.ladderize` a logical indicating whether the tree should be (right) ladderized.

`tree.type` a character string specifying the type of phylogeny to be drawn. Can be `"phylogram"`, `"cladogram"` or `"fan"`.

`tree.ratio` a numeric value in [0, 1] giving the proportion of width of the figure for the tree.

`tree.xlim` a numeric vector of length 2 giving the limits of the x-axis for the tree. If `NULL`, it is determined automatically.

`tree.open.angle`
a numeric value giving the angle in degrees left blank if `tree.type = "fan"`.

`tree.open.crown`
a logical indicating whether the crowns should be drawn following the value of `tree.open.angle` (default `TRUE`).

`show.tip` logical indicating whether tips labels should be drawn.

`tip.labels` character vector to label the tips. If `NULL` the tips labels of the `phylo4d` object are used

`tip.col` a vector of R colors to use for the tips labels. Recycled if necessary.

`tip.cex` a numeric vector to control character size of the tips labels. Recycled if necessary.

`tip.font` an integer vector specifying the type of font for the tips labels: 1 (plain text), 2 (bold), 3 (italic), or 4 (bold italic). Recycled if necessary.

`tip.adj` a vector of numeric in [0, 1] to control tips labels justification: 0 (left-justification), 0.5 (centering), or 1 (right-justification). Recycled if necessary.

`cell.col` a vector of colors for `gridplot` cells. Easily generated by `heat.colors`, `topo.colors`, `terrain.colors` or other functions created with `colorRampPalette`.

`show.color.scale`
logical indicating whether color scale should be drawn.

`show.trait` logical indicating whether traits labels should be drawn.

`trait.labels` character vector to label the traits. If `NULL` the traits labels of the `phylo4d` object are used.

`trait.col` a vector of R colors to use for the traits labels. Recycled if necessary.

`trait.cex` a numeric vector to control character size of the trait labels. Recycled if necessary.

`trait.font` an integer vector specifying the type of font for the traits labels: 1 (plain text), 2 (bold), 3 (italic), or 4 (bold italic). Recycled if necessary.

`trait.bg.col` a vector of R colors to use for the background of the barplots. Recycled if necessary.

`show.box` a logical indicating whether a box should be drawn around the plots.

`grid.vertical` a logical incating whether vertical lines of the grid should be drawn.

`grid.horizontal`
a logical incating whether horizontal lines of the grid should be drawn.

`grid.col` a vector of R colors to use for the lines of the grid.

`grid.lty` the lines type of the grid. Possibly a vector.

`...` further arguments to be passed to `plot.phylo`.

Examples

```
data(navic)
gridplot(navic)

# Multivariate data
require(phylobase)
tipData(navic) <- matrix(rnorm(170), nrow = 17)
gridplot(navic)
```

`kStarTest` *Computes permutation test for Blomberg's K Star*

Description

Computes permutation test for Blomberg's K Star

Usage

```
kStarTest(xr, vcvr, reps)
```

Arguments

`xr` a vector of data

`vcvr` phylogenetic variance-covariance matrix

`reps` the number of permutations

kTest *Computes permutation test for Blomberg's K*

Description

Computes permutation test for Blomberg's K

Usage

```
kTest(xr, vcvr, reps)
```

Arguments

xr	a vector of data
vcvr	phylogenetic variance-covariance matrix
reps	the number of permutations

lambdaTest *Test Pagel's Lambda Optimize Pagel's Lambda and do a likelihood ratio test.*

Description

Test Pagel's Lambda Optimize Pagel's Lambda and do a likelihood ratio test.

Usage

```
lambdaTest(x, vcv)
```

Arguments

x	a vector of numeric data.
vcv	the phylogenetic variance-covariance matrix.

Details

The optimization process is currently performed in R.

`lipaMoran` *Local Indicator of Phylogenetic Association*

Description

This function computes Local Indicator of Phylogenetic Association (local Moran's I) for each tip of a tree. Tests are based on permutations.

Usage

```
lipaMoran(
  p4d,
  trait = names(tdata(p4d)),
  reps = 999,
  alternative = "greater",
  prox.phylo = "patristic",
  as.p4d = FALSE
)
```

Arguments

`p4d`	a `phylo4d` object.
`trait`	the traits in the `phylo4d` object for which to compute LIPA. Can be a character vector giving the name of the traits or numbers giving the column in the table of the data slot of the `phylo4d` object.
`reps`	a numeric value. Number of repetitions for the estimation of p.values with randomization.
`alternative`	a character string specifying the alternative hypothesis for the tests. Must be one of `greater` (default), `two-sided` or `less`.
`prox.phylo`	a matrix of phylogenetic proximities or a character string specifying a method to compute it. See Details.
`as.p4d`	logical. Should the results returned as a `phylo4d` object?

Details

If "`prox.phylo`" is a character string, the phylogenetic proximity matrix is computed internally using the function `proxTips` from the package **adephylo**. Different methods are available: "`patristic`","`nNodes`","`Abouheif`" and "`sumDD`". See `proxTips` for details about the methods.

Value

If `as.p4d` is `FALSE` (default), the function returns a list:

lipa A matrix of LIPA indices computed for each tip of the tree and each trait.

p.value A matrix of p-values (tests of LIPA indices)

reps Number of permutations for the tests

alternative Alternative hypothesis for the tests

If `as.p4d` is `TRUE`, the function returns a `phylo4d` object with LIPA values as tips associated data.

References

Anselin L. (1995) Local Indicators of Spatial Association-LISA. Geographical Analysis 27, 93-115.

`mantelStat` *Mantel statistic*

Description

Mantel statistic

Usage

```
mantelStat(xr, Wr)
```

Arguments

`xr`	a matrix of traits.
`Wr`	a matrix of weights.

`matchTipsAndTraits` *Match matrix row/col names with phylo4d tips/traits*

Description

Check for constitency of rows and columns names between a matrix of traits values and phylo4d object.

Usage

```
matchTipsAndTraits(
  x,
  p4d = NULL,
  p4d.tips = NULL,
  p4d.traits = NULL,
  subset = TRUE
)
```

Arguments

`x`	a matrix of data.
`p4d`	a phylo4d object.
`p4d.tips`	tips labels (relevant if p4d is `NULL`).
`p4d.traits`	traits labels (relevant if p4d is `NULL`).
`subset`	a logical. Should the data matrix be subsetted using tips and traits labels.

Value

The data matrix (eventually subsetted). An error if no consistency between the data and the tree.

`moranTest` *Computes permutation test for Moran's I*

Description

Computes permutation test for Moran's I

Usage

```
moranTest(xr, Wr, reps)
```

Arguments

`xr`	a vector of data
`Wr`	a weighting matrix
`reps`	the number of permutations

`multiplot.phylo4d` *Plots of Traits Values along a Phylogeny*

Description

This function provides a general interface to plot `phylo4d` object (i.e. phylogenetic tree and data).

Usage

```
multiplot.phylo4d(
  p4d,
  trait = names(tdata(p4d)),
  center = TRUE,
  scale = TRUE,
  plot.type = "barplot",
  tree.ladderize = FALSE,
  tree.type = "phylogram",
  tree.ratio = NULL,
  tree.xlim = NULL,
  tree.open.angle = 0,
  tree.open.crown = TRUE,
  show.tip = TRUE,
  tip.labels = NULL,
  tip.col = "black",
  tip.cex = 1,
  tip.font = 3,
  tip.adj = 0,
  data.xlim = NULL,
  bar.lwd = 10,
  bar.col = "grey35",
  show.data.axis = TRUE,
  dot.col = "black",
  dot.pch = 20,
  dot.cex = 2,
  cell.col = white2red(100),
  show.color.scale = TRUE,
  show.trait = TRUE,
  trait.labels = NULL,
  trait.col = "black",
  trait.cex = 1,
  trait.font = 1,
  trait.bg.col = "grey90",
  error.bar.sup = NULL,
  error.bar.inf = NULL,
  error.bar.col = 1,
  show.box = FALSE,
  grid.vertical = TRUE,
  grid.horizontal = FALSE,
  grid.col = "grey25",
  grid.lty = "dashed",
  ...
)
```

Arguments

p4d a phylo4d object.

`trait` the traits in the `phylo4d` object to include in the plot. Can be a character vector giving the name of the traits or numbers giving the column index in the table of the data slot of the p4d object. Can be used to reorder the traits in the plot.

`center` a logical indicating whether traits values should be centered.

`scale` a logical indicating whether traits values should be scaled.

`plot.type` a character string specifying the type of plot for traits data. Can be `"barplot"`, `"dotplot"` or `"gridplot"`.

`tree.ladderize` a logical indicating whether the tree should be (right) ladderized.

`tree.type` a character string specifying the type of phylogeny to be drawn. Can be `"phylogram"`, `"cladogram"` or `"fan"`.

`tree.ratio` a numeric value in [0, 1] giving the proportion of width of the figure for the tree.

`tree.xlim` a numeric vector of length 2 giving the limits of the x-axis for the tree. If `NULL`, it is determined automatically.

`tree.open.angle`
a numeric value giving the angle in degrees left blank if `tree.type = "fan"`.

`tree.open.crown`
a logical indicating whether the crowns should be drawn following the value of `tree.open.angle` (default `TRUE`).

`show.tip` logical indicating whether tips labels should be drawn.

`tip.labels` character vector to label the tips. If `NULL` the tips labels of the `phylo4d` object are used

`tip.col` a vector of R colors to use for the tips labels. Recycled if necessary.

`tip.cex` a numeric vector to control character size of the tips labels. Recycled if necessary.

`tip.font` an integer vector specifying the type of font for the tips labels: 1 (plain text), 2 (bold), 3 (italic), or 4 (bold italic). Recycled if necessary.

`tip.adj` a vector of numeric in [0, 1] to control tips labels justification: 0 (left-justification), 0.5 (centering), or 1 (right-justification). Recycled if necessary.

`data.xlim` numeric vector of length 2 or matrix giving the x coordinates range for the barplots/dotplots (see Details).

`bar.lwd` a vector of numeric giving bar widths of the barplot(s). Recycled along the tips, reapeated for each trait.

`bar.col` a vector of R colors to use for the bars. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning (see Details)

`show.data.axis` logical indicating whether barplots/dotplots axes should be drawn.

`dot.col` a vector of R colors to use for the points. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning (see Details)

`dot.pch` a numerical vector of symbol to use for the points. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning (see Details)

`dot.cex` a numerical vector. Character (or symbol) expansion for the points. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning (see Details) `cell.col` a vector of colors for `gridplot` cells. Easily generated by `heat.colors`, `topo.colors`, `terrain.colors` or other functions created with `colorRampPalette`.

`show.color.scale` logical indicating whether color scale should be drawn.

`show.trait` logical indicating whether traits labels should be drawn.

`trait.labels` character vector to label the traits. If `NULL` the traits labels of the `phylo4d` object are used.

`trait.col` a vector of R colors to use for the traits labels. Recycled if necessary.

`trait.cex` a numeric vector to control character size of the trait labels. Recycled if necessary.

`trait.font` an integer vector specifying the type of font for the traits labels: 1 (plain text), 2 (bold), 3 (italic), or 4 (bold italic). Recycled if necessary.

`trait.bg.col` a vector of R colors to use for the background of the barplots. Recycled if necessary.

`error.bar.sup` a matrix giving the superior limit for error bars. Columns and rows names must match with traits and tips labels, respectively.

`error.bar.inf` a matrix giving the inferior limit for error bars. Columns and rows names must match with traits and tips labels, respectively.

`error.bar.col` a vector of R colors to use for the bars. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning (see Details)

`show.box` a logical indicating whether a box should be drawn around the plots.

`grid.vertical` a logical incating whether vertical lines of the grid should be drawn.

`grid.horizontal` a logical incating whether horizontal lines of the grid should be drawn.

`grid.col` a vector of R colors to use for the lines of the grid.

`grid.lty` the lines type of the grid. Possibly a vector.

`...` further arguments to be passed to `plot.phylo`.

---

navic *Phylogeny and pollution sensitivity of diatoms*

Description

A phylogenetic tree and the pollution sensitivity of 17 diatoms species from the order Naviculales.

Usage

```
data(navic)
```

Format

a `phylo4d` object.

Source

Keck F., Rimet F., Franc A. & Bouchez A. (In press) Phylogenetic signal in diatom ecology: perspectives for aquatic ecosystems biomonitoring. Ecological Applications.

`pagelLogLik` *Computes log-likelihood for data and a given value of Pagel's Lambda*

Description

Computes log-likelihood for data and a given value of Pagel's Lambda

Usage

```
pagelLogLik(lambda, xr, vcvr)
```

Arguments

`lambda`	the value of Pagel's Lambda
`xr`	a vector of data
`vcvr`	phylogenetic variance-covariance matrix

Author(s)

Adapted from Liam Revell's R function 'phylosig' in phytools.

`phyloCorrelogram` *Phylogenetic correlogram*

Description

This function computes a phylogenetic correlogram.

Usage

```
phyloCorrelogram(
  p4d,
  trait = names(tdata(p4d)),
  dist.phylo = "patristic",
  sigma = NULL,
  n.points = 100,
  ci.bs = 1000,
  ci.conf = 0.95
)
```

Arguments

`p4d` a `phylo4d` object.

`trait` the traits in the `phylo4d` object to use for the correlogram. Can be a character vector giving the name of the traits or numbers giving the column index in the table of the data slot of the `phylo4d` object.

`dist.phylo` a matrix of phylogenetic distances or a character string specifying a method to compute it. See Details.

`sigma` a numeric value giving the standard deviation of the normal distribution used to compute the matrix of phylogenetic weights. If `NULL` (default), the function computes a value from the phylogeny.

`n.points` an integer giving the number of points at which to compute the correlogram's statistics.

`ci.bs` an integer giving the number of bootstrap replicates for confidence interval estimation.

`ci.conf` a value between 0 and 1 giving the confidence level of the confidence interval.

Details

This function computes a correlogram on a continuous scale of phylogenetic distance. This is achieved by using a collection of specific phylogenetic weights matrices generated with the "`lag-norm`" method of `phyloWeights` and different values of "`mu`".

The confidence envelope is computed by bootstrapping. At each iteration, the autocorrelation is re-estimated after re-standardization of the matrix of phylogenetic weights. The nonparametric confidence intervals are computed at each lag by first order normal approximation. Intervals are constrained between 0 and 1.

If there is one trait, the function computes Moran's I. If there is more than one trait, the function computes the Mantel's statistic (Oden and Sokal 1986).

If "`dist.phylo`" is a character string, the phylogenetic distance matrix is computed internally using the function `distTips` from the package **adephylo**. See `distTips` for details about the methods.

Value

An object of class "`phylocorrelogram`".

References

Oden N.L. & Sokal R.R. (1986) Directional Autocorrelation: An Extension of Spatial Correlograms to Two Dimensions. Systematic Zoology 35, 608-617.

See Also

`plot.phylocorrelogram`, `correlogram.formula` in **ape** for correlograms based on taxonomic levels.

Examples

```
## Not run:
data(navic)
pc <- phyloCorrelogram(navic)
plot(pc)

## End(Not run)
```

---

phyloSignal *Computes phylogenetic signal with different methods*

Description

This function computes phylogenetic signal statistics (Blomberg's K and K*, Abouheif's Cmean, Moran's I, and Pagel's Lambda) for traits in `phylo4d` objects.

Usage

```
phyloSignal(
  p4d,
  methods = c("all", "I", "Cmean", "Lambda", "K", "K.star"),
  reps = 999,
  W = NULL
)
```

Arguments

`p4d`	a `phylo4d` object.
`methods`	a character vector giving the methods to compute phylogenetic signal (see Details).
`reps`	an integer. The number of repetitions for the estimation of p.values with randomization.
`W`	an optional matrix of phylogenetic weights to compute Moran's I. By default the matrix is computed with the function `proxTips` with patristic distances.

Details

p4d must be a `phylo4d` object as defined in **phylobase** package. By default, the `methods` argument is set to "`all`" and all the available methods are used. The user can specify which method(s) to use. Possible methods are "`I`" (Gittleman & Kot 1990), "`Cmean`" (Abouheif 1999)", "`Lambda`" (Pagel 1999), "`K`" and "`K.star`" (Blomberg et al. 2003).

Value

A list of two dataframes with the values of statistics and associated p.values for each tested trait and method.

Author(s)

This function is a general wrapper for C++ subroutines. C++ code is adapted from R functions in Pavoine and Ricotta (2013) and Revell (2012).

References

Abouheif E. (1999) A method for testing the assumption of phylogenetic independence in comparative data. Evolutionary Ecology Research 1, 895-909. Blomberg S.P., Garland Jr T. & Ives A.R. (2003) Testing for phylogenetic signal in comparative data: behavioral traits are more labile. Evolution 57, 717-745. Gittleman J.L. & Kot M. (1990) Adaptation: Statistics and a null model for estimating phylogenetic effects. Systematic Biology 39, 227-241. Pagel M. (1999) Inferring the historical patterns of biological evolution. Nature 401, 877-884. Revell L.J. (2012) phytools: an R package for phylogenetic comparative biology (and other things). Methods in Ecology and Evolution 3, 217-223. Pavoine S. & Ricotta C. (2013) Testing for Phylogenetic Signal in Biological Traits: The Ubiquity of Cross-Product Statistics. Evolution 67, 828-840.

See Also

`phyloSimSignal` .

Examples

```
require(ape)
require(phylobase)
data(navic)
tipData(navic)$rand <- rnorm(17)
tipData(navic)$BM <- rTraitCont(as(navic, "phylo"))
phyloSignal(navic)
```

`phylosignal` *phylosignal*

Description

A collection of tools to explore the phylogenetic signal in univariate and multivariate data. The package provides functions to plot traits data against a phylogenetic tree, different measures and tests for the phylogenetic signal, methods to describe where the signal is located and a phylogenetic clustering method.

phyloSignalBS *Computes phylogenetic signal for bootstrapped replicates of a phylogeny.*

Description

This function computes phylogenetic signal statistics and p-values for bootsrapped replicates a phylogenetic tree and produce boxplots to represent the results. This can be useful to check the impact of phylogenetic reconstruction uncertainty on phylogenetic signal.

Usage

```
phyloSignalBS(
  p4d,
  multiphylo,
  methods = c("all", "I", "Cmean", "Lambda", "K", "K.star"),
  reps = 999,
  W = NULL,
  pb = TRUE
)
```

Arguments

`p4d`	a `phylo4d` object.
`multiphylo`	a `multiphylo` object containing bootstrapped trees of `p4d`.
`methods`	a character vector giving the methods to compute phylogenetic signal (see `phyloSignal`).
`reps`	an integer. The number of repetitions for the estimation of p.values with randomization.
`W`	an optional matrix of phylogenetic weights to compute Moran's I. By default the matrix is computed with the function `proxTips` with patristic distances.
`pb`	a logical. Should a progress bar be printed? (default `TRUE`).

Details

Time consumption can be important if there are many bootraped trees and tested traits.

Value

The data generated are returned invisibly as a list.

phyloSignalINT *Computes phylogenetic signal at each internal node of a phylogeny*

Description

This function computes phylogenetic signal statistics and p-values for a given trait and a given method at each internal node of a phylogenetic tree.

Usage

```
phyloSignalINT(
  p4d,
  trait = names(tipData(p4d))[1],
  method = "Cmean",
  reps = 999,
  W = NULL
)
```

Arguments

`p4d`	a `phylo4d` object.
`trait`	a character string giving the trait to use to compute the signal. By default the first trait is taken from `p4d`.
`method`	a character vector giving the method to use to compute phylogenetic signal (default is `"Cmean"`; see `phyloSignal`).
`reps`	an integer. The number of repetitions for the estimation of p.values with randomization.
`W`	an optional matrix of phylogenetic weights to compute Moran's I. By default the matrix is computed with the function `proxTips` with patristic distances.

Value

A `phylo4d` object with phylogenetic signal statistics and p-values as nodes associated data.

phyloSim *Simulate the behaviour of phylogenetic signal statistics with a given phylogeny*

Description

This function simulates different phylogenetic signal statistics for a given phylogenetic tree along a gradient of Brownian Motion influence.

Usage

```
phyloSim(
  tree,
  methods = c("all", "I", "Cmean", "Lambda", "K", "K.star"),
  nsim = 99,
  reps = 999,
  W = NULL,
  model = "BM",
  pb = TRUE
)
```

Arguments

`tree`	a `phylo`, `phylo4` or `phylo4d` object.
`methods`	a character vector giving the methods to compute phylogenetic signal (see Details).
`nsim`	a numeric value. Number of simulated traits at each step in the gradient.
`reps`	a numeric value. Number of repetitions for the estimation of p.values with randomization.
`W`	an optional matrix of phylogenetic weights to compute Moran's I. By default the matrix is computed with the function `proxTips` with patristic distances.
`model`	the model to use for traits simulation (only `"BM"`, default, is available).
`pb`	a logical. Should a progress bar be printed? (default `TRUE`).

Details

By default, the `methods` argument is set to `"all"` and all the available methods are used. The user can specify which method(s) to use. Possible values are `"I"`, `"Cmean"`, `"Lambda"`, `"K"` and `"K.star"`, see `phyloSignal` for further details.

Value

An object of class `phylosim`.

See Also

`phyloSimSignal`.

Examples

```
## Not run:
data(navic)
psim <- phyloSim(navic)
plot(psim)
plot.phylosim(psim, what = "pval", stacked.methods = TRUE)

## End(Not run)
```

phyloSimSignal *Phylogenetic signal estimation as a fraction of a Brownian Motion process.*

Description

This function crosses traits data with simulations to estimate the phylogenetic signal as a fraction of a Brownian Motion process. This is experimental.

Usage

```
phyloSimSignal(p4d, phylosim, quantiles = c(0.05, 0.95))
```

Arguments

`p4d`	a `phylo4d` object.
`phylosim`	a `phylosim` object.
`quantiles`	a vector of two numeric values between 0 and 1 giving the minimum and the maximum quantiles to estimate statistics fluctuations.

Value

An object of class `phylosimsignal` with a `print` and a `plot` method.

See Also

`phyloSignal`, `phyloSim`.

phyloWeights *Phylogenetic weights matrix*

Description

This function can be used to compute a phylogenetic weights matrix with different methods.

Usage

```
phyloWeights(
  tree,
  dist.phylo = "patristic",
  method = "lag-norm",
  mu = 0,
  sigma = 5,
  dmax = 10,
  alpha = 1,
  beta = 1
)
```

Arguments

tree	a phylo, phylo4 or phylo4d object.
dist.phylo	a character string specifying the method used to compute phylogenetic distances. Available distances are "patristic","nNodes","Abouheif" and "sumDD". See Details.
method	a method to compute phylogenetic weights from phylogenetic distances. Available methods are "lag-norm", "clade", "inverse" and "exponential". See Details.
mu	a numeric value giving the mean of the distribution if method is lag-norm. This is a phylogenetic distance.
sigma	a numeric value giving the standard deviation of the distribution if method is lag-norm.
dmax	the maximum phylogenetic distance to use to delineate clades.
alpha	a numeric value giving the exponent to use if method is inverse.
beta	a numeric value giving the factor to use if method is exponential.

Details

Method "inverse":

$$\frac{1}{d^{\alpha}}$$

The phylogenetic distance matrix is computed internally using the function distTips from the package **adephylo**. See distTips for details about the methods.

Value

A square matrix of phylogenetic weights whose sums of rows is 1.

See Also

proxTips in **adephylo**.

plot.graphclust *Plot phylogenetically constrained clustering*

Description

This function produces three plots (selectable by which): a plot of edges selection based on phylogenetic against trait distances of taxa pairs, a plot of the graph produced with the selected edges and a plot of the clustered phylogenetic tree.

Usage

```
## S3 method for class 'graphclust'
plot(
  x,
  which = c("selection", "graph", "tree"),
  ask = TRUE,
  colored = TRUE,
  ...
)
```

Arguments

x	a graphclust object as produced by graphClust.
which	a character vector to select plots. Must be one or more of "selection", "graph", "tree".
ask	logical if TRUE (default), the user is asked before each plot.
colored	logical indicating if plots must include colors.
...	further arguments to be passed to or from other methods. They are currently ignored in this function.

Examples

```
data(navic)
gC <- graphClust(navic, lim.phylo = 1, lim.trait = 2, scale.lim = FALSE)
plot.graphclust(gC, which = "selection", ask = FALSE)
plot.graphclust(gC, which = "graph", ask = FALSE)
plot.graphclust(gC, which = "tree", ask = FALSE)
```

plot.phylocorrelogram *Plot a phylogenetic correlogram*

Description

This function plots phylogenetic correlograms produced by phyloCorrelogram

Usage

```
## S3 method for class 'phylocorrelogram'
plot(
  x,
  show.ci = TRUE,
  show.h0 = TRUE,
  show.test = TRUE,
  xlab = "Phylogenetic distance",
  ylab = "Correlation",
  main = "Phylogenetic correlogram",
  ...
)
```

Arguments

x	a phylocorrelogram object.
show.ci	a logical indicating whether to plot the confidence interval envelop (default TRUE).
show.h0	a logical indicating whether to plot the the line showing the expected value of Moran's I under the null hypothesis of no phylogenetic autocorrelation (default TRUE).
show.test	a logical indicating whether to plot indicator of significance (default TRUE).
xlab	a label for the x axis.
ylab	a label for the y axis.
main	a main title for the plot
...	other graphical parameters passed to the plot function.

Examples

```
## Not run:
data(navic)
pc <- phyloCorrelogram(navic)
plot(pc)

## End(Not run)
```

plot.phylosim *Plot* phylosim *object*

Description

This function plots a phylosim object to visualize the behaviour of phylogenetic signal statistics for a given phylogenetic tree

Usage

```
## S3 method for class 'phylosim'
plot(
  x,
  what = c("stat", "pval"),
  stacked.methods = FALSE,
  quantiles = c(0.05, 0.95),
  col = 1:5,
  legend = TRUE,
  ...
)
```

Arguments

`x` a `phylosim` object.

`what` what to represent on the plot. Can be the statistics used to measure the signal ("`stat`") or the p-values ("`pval`").

`stacked.methods` a logical. If different methods have been used, should they be plotted on the same graphic (`TRUE`) or not (`FALSE`, default).

`quantiles` a vector of two numeric values between 0 and 1 giving the minimum and the maximum quantiles to plot. Set to `NULL` to not plot quantiles.

`col` a vector of colors for the different methods.

`legend` a logical. If `stacked.methods` is set to `TRUE`, should a legend be printed to differentiate the different methods?

`...` further arguments to be passed to or from other methods.

See Also

`phyloSim`.

Examples

```
## Not run:
data(navic)
psim <- phyloSim(navic)
plot(psim)
plot.phylosim(psim, what = "pval", stacked.methods = TRUE)

## End(Not run)
```

`plot.phylosimsignal` *Plot signal estimation as a fraction of a Brownian Motion process.*

Description

Plot signal estimation as a fraction of a Brownian Motion process.

Usage

```
## S3 method for class 'phylosimsignal'
plot(
  x,
  methods = NULL,
  traits = NULL,
  stacked.methods = FALSE,
  stacked.traits = FALSE,
  print.quantiles = TRUE,
  col = 1:5,
  legend = TRUE,
  ...
)
```

Arguments

x
: an object of class `phylosimsignal`.

methods
: a character vector giving the methods (included in the `phylosimsignal` object) to plot.

traits
: a character vector giving the traits (included in the `phylosimsignal` object) to plot.

stacked.methods
: If different methods have been used, should they be plotted on the same graphic (`TRUE`) or not (`FALSE`, default).

stacked.traits
: If different traits have been used, should they be plotted on the same graphic (`TRUE`) or not (`FALSE`, default).

print.quantiles
: logical stating whether quantiles should be plotted.

col
: a vector of colors for the different methods.

legend
: a logical. If `stacked.methods` is set to `TRUE`, should a legend be printed to differentiate the different methods?

...
: further arguments to be passed to or from other methods.

print.graphclust *#' Print results of phylogenetically constrained clustering*

Description

#' Print results of phylogenetically constrained clustering

Usage

```
## S3 method for class 'graphclust'
print(x, ...)
```

Arguments

x
: a `graphclust` object as produced by `graphClust`.

...
: further arguments to be passed to or from other methods.

print.phylosimsignal *Print signal estimation as a fraction of a Brownian Motion process.*

Description

Print signal estimation as a fraction of a Brownian Motion process.

Usage

```
## S3 method for class 'phylosimsignal'
print(x, ...)
```

Arguments

`x`	an object of class `phylosimsignal`.
`...`	further arguments to be passed to or from other methods.

read.p4d *Read a phylo4d object from files*

Description

This function creates an object of class `phylo4d` by combining a phylogenetic tree and its associated tips data stored in two distinct files.

Usage

```
read.p4d(phylo.file, data.file, phylo.format = "newick", data.format = "table")
```

Arguments

`phylo.file`	the name of the file which the phylogenetic tree is to be read from.
`data.file`	the name of the file which the tips data are to be read from.
`phylo.format`	the format of the phylogenetic tree provided. Possible formats are "`newick`" and "`nexus`".
`data.format`	the format of the table with tips data. Possible formats are "`table`", "`csv`", "`csv2`", "`delim`" and "`delim2`".

Details

phylo.file and data.file can be provided as objects of mode character or double-quoted strings.

The phylogenetic tree can be imported in two formats.

- The newick format refers to the simple parenthetic format known as the Newick or New Hampshire format. The tree is read by calling the function read.tree of the **ape** package.
- The nexus format refers to NEXUS format. The tree is read by calling the function read.nexus of the **ape** package.

Tips data are imported from a table formatted file. The different formats allow to use different separator and decimal characters. They correspond to the variants of read.table:

- table use read.table with default settings.
- csv use read.csv with default settings.
- csv2 use read.csv2 with default settings.
- delim use read.delim with default settings.
- delim2 use read.delim2 with default settings.

Value

An object of class phylo4d.

See Also

phylo4d to create a phylo4d object.

---

rTraitContWeight *Traits Values Simulation*

Description

This function simulates the evolution of continuous characters along a phylogeny. Traits values can be more or less influenced by the provided model.

Usage

```
rTraitContWeight(tree, model = "BM", weight = 1, as.p4d = FALSE)
```

Arguments

tree	an object of class "phylo.
model	the model to use ("BM" or "OU").
weight	a numeric vector with values ranging between 0 and 1 giving the balance between pure random evolution and the selected model.
as.p4d	logical. Should the phylogenetic tree and the simulated data coerced to a phylo4d object?

Value

An object of class phyloSimSignal.

`subsetPhyloSimSignal` *phyloSimSignal subsetting*

Description

This function subset all the components of a phyloSimSignal object following the methods and traits provided.

Usage

```
subsetPhyloSimSignal(x, methods, traits)
```

Arguments

`x`	an object of class `phyloSimSignal` to subset.
`methods`	a character vector giving the methods to keep.
`traits`	a character vector giving the traits to keep.

Value

A `phyloSimSignal` object

`white2red` *Color Palette*

Description

A simple color palette for gridplots.

Usage

```
white2red(n)
```

Arguments

`n`	the number of colors to be in the palette.

Index