Package ‘phyloregion’

July 15, 2025

Type Package

Title Biogeographic Regionalization and Macroecology

Version 1.0.9

Description Computational infrastructure for biogeography, community ecology, and biodiversity conservation (Daru et al. 2020) <doi:10.1111/2041-210X.13478>. It is based on the methods described in Daru et al. (2020) <doi:10.1038/s41467-020-15921-6>. The original conceptual work is described in Daru et al. (2017) <doi:10.1016/j.tree.2017.08.013> on patterns and processes of biogeographical regionalization. Additionally, the package contains fast and efficient functions to compute more standard conservation measures such as phylogenetic diversity, phylogenetic endemism, evolutionary distinctiveness and global endangerment, as well as compositional turnover (e.g., beta diversity).

Imports ape, phangorn, Matrix, betapart, parallel, methods, colorspace, igraph, clustMixType, maptpx, terra, vegan, predicts, smoothr

Suggests tinytest, knitr, rmarkdown, mapproj, survival, rJava, phyloseq, V8

VignetteBuilder knitr

URL `https://github.com/darunabas/phyloregion`, `https://phyloregion.com/index.html`

BugReports `https://github.com/darunabas/phyloregion/issues`

License AGPL-3

Encoding UTF-8

RoxygenNote 7.3.2

biocViews Software

NeedsCompilation no

Depends R (>= 4.1.0)

Author Barnabas H. Daru [aut, cre, cph] (ORCID: <`https://orcid.org/0000-0002-2115-0257`>), Piyal Karunarathne [aut] (ORCID: <`https://orcid.org/0000-0002-1934-145X`>),

Klaus Schliep [aut] (ORCID: <https://orcid.org/0000-0003-2941-0161>),
Xiaobei Zhao [ctb],
Albin Sandelin [ctb],
Luciano Pataro [ctb]

Maintainer Barnabas H. Daru <darunabas@gmail.com>

Repository CRAN

Date/Publication 2025-07-15 06:20:02 UTC

Contents

africa 3
arc_labels 4
backg 6
beta_core 6
choropleth 7
coldspots 8
collapse_range 9
counts 10
dirpath 11
EDGE 12
evol_distinct 13
fishnet 14
fitgom 15
functional_beta 17
get_clades 19
hexcols 19
indicators 20
long2sparse 21
map_trait 22
match_phylo_comm 23
mean_dist 24
nodepie 24
optimal_phyloregion 25
PD 26
PD_ses 27
phylobeta_core 29
phylobeta_ses 30
phylobuilder 31
phyloregion 32
phylo_endemism 34
plot.phyloregion 35
plot.sparse 36
plot_spatial_membership 37
random_species 38
rast2comm 38
rast_quantile 40
read.community 40

sdm 41
selectbylocation 43
select_linkage 44
timeslice 44
tree_sampler 45
unifrac 46
weighted_endemism 47

Index **48**

`africa` *Plants of southern Africa*

Description

This dataset consists of a dated phylogeny of the woody plant species of southern Africa along with their geographical distributions. The dataset comes from a study that maps tree diversity hotspots in southern Africa (Daru et al. 2015). The study mapped five types of diversity hotspots including species richness (SR), phylogenetic diversity (PD), phylogenetic endemism (PE), species weighted endemism (CWE), and evolutionary distinctiveness and global endangerment (EDGE). The results revealed large spatial incongruence between biodiversity indices, resulting in unequal representation of PD, SR, PE, CWE and EDGE in hotspots and currently protected areas, suggesting that an integrative approach which considers multiple facets of biodiversity is needed to maximise the conservation of tree diversity in southern Africa. Specifically for this package, we arranged the dataset into four components: “comm”, “polys”, “phylo”, “mat”, “IUCN”.

Details

- comm: This is a sparse community composition matrix of each species presences/absences within 50 × 50 km grid cells. A sparse matrix is a matrix with a high proportion of zero entries (Duff 1977), of which only the non-zero entries are stored and used for downstream analysis.
- polys: These are the grid cells covering the geographic extent of study area. These can be created using the function `fishnet`. The polys object is of class `SpatVector` and has a column labeled “grids”, with the grid identities.
- phylo: This corresponds to the phylogenetic tree which was estimated using Bayesian analysis of 1,400 species and 1,633 bp of chloroplast DNA sequences derived from a combination of *matK* and *rbcLa*, assuming an uncorrelated relaxed molecular clock model, using the program BEAST v.1.7.5 (Drummond & Rambaut, 2007). Branch lengths were calibrated in millions of years using a Bayesian MCMC approach by enforcing topological constraints assuming APG III backbone from Phylomatic v.3 (Webb & Donoghue, 2005) and 18 fossil calibration points from Bell et al. (2010).
- mat: This is a distance matrix of phylogenetic beta diversity between all grid cells at the 50 × 50 km scale.
- IUCN: This is a dataframe of IUCN conservation status of each woody species (LC, NT, VU, EN, CR). This is useful for analysis of Evolutionary Distinctiveness and Global Endangerment using the function `EDGE`.

References

Bell, C.D., Soltis, D.E., & Soltis, P.S. (2010). The age and diversification of the angiosperms re-revisited. *American Journal of Botany* **97**, 1296–1303.

Daru, B.H., Van der Bank, M. & Davies, T.J. (2015) Spatial incongruence among hotspots and complementary areas of tree diversity in southern Africa. *Diversity and Distributions* **21**, 769-780.

Drummond, A.J., & Rambaut, A. (2007). BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evolutionary Biology* **7**, 214.

Duff, I.S. (1977). A survey of sparse matrix research. *Proceedings of the IEEE* **65**, 500–535.

Webb, C.O., & Donoghue, M.J. (2005). Phylomatic: Tree assembly for applied phylogenetics. *Molecular Ecology Notes* **5**, 181–183.

Examples

```
data(africa)
names(africa)

library(terra)
library(ape)
plot(africa$phylo)
```

`arc_labels` *Add arc labels to plotted phylogeny*

Description

Add arc labels to plotted phylogeny

Usage

```
arc_labels(phy, tips, ...)

## Default S3 method:
arc_labels(
  phy = NULL,
  tips,
  text,
  plot_singletons = TRUE,
  ln.offset = 1.02,
  lab.offset = 1.06,
  cex = 1,
  orientation = "horizontal",
  ...
)
```

Arguments

`phy`	An object of class phylo.
`tips`	A character vector (or a list) with names of the tips that belong to the clade or group. If multiple groups are to be plotted, tips must be given in the form of a list.
`...`	Further arguments passed to or from other methods.
`text`	Desired clade label.
`plot_singletons`	Logical. If TRUE (default), adds arcs (and labels) to single tip lineages. If FALSE, no arc or labels will be plotted over that tip..
`ln.offset`	Line offset (as a function of total tree height)
`lab.offset`	Label offset.
`cex`	Character expansion
`orientation`	Orientation of the text. Can be “vertical”, “horizontal”, or “curved”.

Value

`NULL`

Examples

```
old.par <- par(no.readonly = TRUE)
require(ape)
data(africa)
par(mai=rep(0,4))
plot(africa$phylo, type = "fan", show.tip.label=FALSE,
     open.angle = 180, edge.width=0.5)

y <- data.frame(species=africa$phylo$tip.label)
y$genus <- gsub("_.*", "\\1", y$species)

fx <- split(y, f=y$genus)

suppressWarnings(invisible(lapply(fx, function(x) {
  y <- seq(from = 1.03, to = 1.09, by = ((1.09 - 1.03)/(length(fx) - 1)))
  z <- sample(y, 1, replace = FALSE, prob = NULL)
  if(nrow(x) > 10L) arc_labels(phy = africa$phylo, tips=x$species,
                             text=as.character(unique(x$genus)),
                             orientation = "curved", cex=0.5,
                             lab.offset = z)
})))
par(old.par)
```

`backg` *Sample background points*

Description

Generates background sample as null for species distribution modeling and other things.

Sample random background points using a vector of probability weights.

Usage

```
backg(calib, spatkde, size = 10000, ...)

randpoints(ras, size, prob = NULL, ...)
```

Arguments

`calib`	A SpatRaster of the species calibration area. If a polygon, convert to SpatRaster using rasterize.
`spatkde`	A weighted or unweighted Gaussian Kernel Density estimate (KDE) for all input occurrence records.
`size`	An positive integer of the number of samples to generate.
`...`	further arguments passed to or from other methods.
`ras`	Input SpatRaster
`prob`	Vector of probability weights for obtaining the points sampled.

Value

A dataframe containing the generated background points.

A dataframe with sampled points.

`beta_core` *Taxonomic (non-phylogenetic) beta diversity*

Description

Data are assumed to be presence / absence (0 / 1) and all values greater zero are assumed to reflect presence.

Usage

```
beta_core(x)

beta_diss(x, index.family = "sorensen")
```

Arguments

`x`	an object of class Matrix, where rows are sites and columns are species.
`index.family`	family of dissimilarity indices, partial match of "sorensen" or "jaccard".

Details

`beta_core` is helper function to compute the basic quantities needed for computing the "sorensen" or "jaccard" index.

Value

`beta_core` returns an object of class `beta_diss` like the `betapart.core` function. This object can be called by `beta.pair` or `beta.multi`.

`beta_diss` returns a list with three dissimilarity matrices. See `beta.pair` for details.

Author(s)

Klaus Schliep

See Also

`betapart.core`, `betapart`, `phylobeta`

Examples

```
data(africa)
x <- africa$comm
bc <- beta_core(x)
beta_sorensen <- beta_diss(x)
```

`choropleth` *Bin values*

Description

`choropleth` discretizes the values of a quantity for mapping.

Usage

```
choropleth(x, k = 10, breaks = "quantile", min = NULL, max = NULL)
```

Arguments

`x`	Vector of values to discretize.
`k`	Numeric, the desired number of bins to discretize.
`breaks`	one of “equal”, “pretty”, “jenks”, “quantile” or numeric vector with the actual breaks by specifying the minimum (`min`) and maximum (`max`) bounds.
`min`	the minima of the lowest bound of the break.
`max`	the maxima of the upper bound of the break

Value

a vector with the discretized values.

Author(s)

Barnabas H. Daru `<darunabas@gmail.com>`

See Also

`coldspots`

`coldspots` *Computes biodiversity coldspots and hotspots*

Description

`coldspots` and `hotspots` map areas or grid cells with lowest or highest values, respectively, of a biodiversity metric e.g. species richness, species endemism or degree of threat.

Usage

```
coldspots(x, y = NULL, prob = 2.5, ...)

hotspots(x, y = NULL, prob = 2.5, ...)

rast_hotspot(ras, ref = NULL, prob = 10)
```

Arguments

`x`	a vector on which to compute hotspots or coldspots
`y`	a vector on which to compare x against
`prob`	The threshold quantile for representing the lowest (`coldspots`) or highest (`hotspots`) proportion of biodiversity in an area. By default, the threshold is set to `prob = 2.5` percent.
`...`	Further arguments passed to or from other methods.
`ras`	a SpatRaster on which to compute hotspots.
`ref`	a raster layer for reference.

Value

A vector of integers of 1s and 0s with 1 corresponding to the coldspots or hotspots

Author(s)

Barnabas H. Daru `<darunabas@gmail.com>`

References

Myers, M., Mittermeier, R.A., Mittermeier, C.G., da Fonseca, G.A.B. & Kent, J. (2000) Biodiversity hotspots for conservation priorities. *Nature* **403**: 853–858.

Ceballos, G. & Ehrlich, P.R. (2006) Global mammal distributions, biodiversity hotspots, and conservation. *Proceedings of the National Academy of Sciences USA* **103**: 19374–19379.

Orme, C.D., Davies, R.G., Burgess, M., Eigenbrod, F., Pickup, N. et al. (2005) Global hotspots of species richness are not congruent with endemism or threat. *Nature* **436**: 1016–1019.

Daru, B.H., Van der Bank, M. & Davies, T.J. (2015) Spatial incongruence among hotspots and complementary areas of tree diversity in southern Africa. *Diversity and Distributions* **21**: 769-780.

Examples

```
library(terra)
data(africa)
p <- vect(system.file("ex/sa.json", package = "phyloregion"))

Endm <- weighted_endemism(africa$comm)
C <- coldspots(Endm, na.rm=TRUE) # coldspots
H <- hotspots(Endm, na.rm=TRUE) # hotspots

## Merge endemism values to shapefile of grid cells.
DF <- data.frame(grids=names(C), cold=C, hot=H)
m <- merge(p, DF, by = "grids", all = TRUE)

plot(p, border = "grey", col = "lightgrey",
     main = "Weighted Endemism Hotspots and Coldspots")
plot(m[(m$cold == 1), ], col = "blue", add = TRUE, border = NA)
plot(m[(m$hot == 1), ], col = "red", add = TRUE, border = NA)
legend("bottomleft", fill = c("blue", "red", "yellow", "green"),
       legend = c("coldspots", "hotspots"), bty = "n", inset = .092)
```

`collapse_range` *Collapse nodes and ranges based on divergence times*

Description

This function collapses nodes and geographic ranges based on species' divergence times at various time depths.

Usage

```
collapse_range(
  x,
  tree,
  n,
  species = "species",
  grids = "grids",
  format = "wide"
)
```

Arguments

`x`	A community matrix or data frame.
`tree`	A phylogenetic tree.
`n`	Time depth to slice the phylogenetic tree (often in millions of years for dated trees).
`species`	If `format` = “long” (the default), the column with the species name.
`grids`	The column with the sites or grids if `format` = “long”.
`format`	Format of the community composition data: “long” or “wide” with species as columns and sites as rows.

Value

Two community data frames: the `collapsed community data` and `original community data`

References

Daru, B.H., Farooq, H., Antonelli, A. & Faurby, S. (2020) Endemism patterns are scale dependent. *Nature Communications* **11**: 2115.

Examples

```
library(ape)
tr1 <- read.tree(text ="(((a:2,(b:1,c:1):1):1,d:3):1,e:4);")
com <- matrix(c(1,0,1,1,0,0,
                1,0,0,1,1,0,
                1,1,1,1,1,1,
                1,0,1,1,0,1,
                0,0,0,1,1,0), 6, 5,
              dimnames=list(paste0("g",1:6), tr1$tip.label))

collapse_range(com, tr1, n=1)
```

`counts` *Phyloregions for functional traits and phylogeny*

Description

Generates a sparse community matrix as input for clustering regions based on the similairity of functional traits across species.

Usage

```
counts(x, trait, cut = NULL, phy = NULL, bin = 10, na.rm = FALSE)
```

Arguments

`x`	A community data in long format with one column representing sites labeled "grids" and another column representing species labeled "species".
`trait`	A data frame or matrix object with the first column labeled "species" containing the taxonomic groups to be evaluated whereas the remaining columns have the various functional traits. The variables must be a mix of numeric and categorical values.
`cut`	The slice time.
`phy`	is a dated phylogenetic tree with branch lengths stored as a phylo object (as in the ape package).
`bin`	The desired number of clusters or bins.
`na.rm`	Logical, whether NA values should be removed or not.

Value

Function returns a community data frame that captures the count of each species based on its cluster membership.

`dirpath` *Get current directory*

Description

Gets the path of the current directory.

Usage

```
dirpath(path)
```

Arguments

`path`	Character vector of the directory path names.

Value

A character vector containing the name of the current directory.

`EDGE` *Evolutionary Distinctiveness and Global Endangerment*

Description

This function calculates EDGE by combining evolutionary distinctiveness (ED; i.e., phylogenetic isolation of a species) with global endangerment (GE) status as defined by the International Union for Conservation of Nature (IUCN).

Usage

```
EDGE(x, phy, Redlist = "Redlist", species = "species", ...)
```

Arguments

`x`	a data.frame
`phy`	a phylogenetic tree (object of class phylo).
`Redlist`	column in the data frame with the IUCN ranks: `LC`, `NT`, `VU`, `EN`, `CR`, and `EX`.
`species`	data frame column specifying the taxon
`...`	Further arguments passed to or from other methods.

Details

EDGE is calculated as:

$$log(1 + ED) + GE * log(2)$$

where *ED* represents the evolutionary distinctiveness score of each species (function `evol_distinct`), i.e. the degree of phylogenetic isolation, and combining it with *GE*, global endangerment from IUCN conservation threat categories. *GE* is calculated as the expected probability of extinction over 100 years of each taxon in the phylogeny (Redding & Mooers, 2006), scaled as follows: least concern = 0.001, near threatened and conservation dependent = 0.01, vulnerable = 0.1, endangered = 0.67, and critically endangered = 0.999.

Value

Returns a dataframe of EDGE scores

Author(s)

Barnabas H. Daru

References

Redding, D.W., & Mooers, A.Ø. (2006) Incorporating evolutionary measures into conservation prioritization. *Conservation Biology* **20**: 1670–1678.

Isaac, N.J., Turvey, S.T., Collen, B., Waterman, C. & Baillie, J.E. (2007) Mammals on the EDGE: conservation priorities based on threat and phylogeny. *PLoS ONE* **2**: e296.

Examples

```
data(africa)
y <- EDGE(x=africa$IUCN, phy=africa$phylo, Redlist="IUCN", species="Species")
```

`evol_distinct` *Species' evolutionary distinctiveness*

Description

Calculates evolutionary distinctiveness measures for a suite of species by: a) equal splits (Redding and Mooers 2006) b) fair proportions (Isaac et al., 2007). This a new implementation of the picante function `evol.distinct` however allowing multifurcations and can be orders of magnitude faster.

Usage

```
evol_distinct(
  tree,
  type = c("equal.splits", "fair.proportion"),
  scale = FALSE,
  use.branch.lengths = TRUE,
  ...
)
```

Arguments

`tree`	an object of class `phylo`.
`type`	a) equal splits (Redding and Mooers 2006) or b) fair proportions (Isaac et al., 2007)
`scale`	The scale option refers to whether or not the phylogeny should be scaled to a depth of 1 or, in the case of an ultrametric tree, scaled such that branch lengths are relative.
`use.branch.lengths`	If use.branch.lengths=FALSE, then all branch lengths are changed to 1.
`...`	Further arguments passed to or from other methods.

Value

a named vector with species scores.

Author(s)

Klaus Schliep

References

Redding, D.W. and Mooers, A.O. (2006). Incorporating evolutionary measures into conservation prioritisation. *Conservation Biology*, **20**, 1670–1678.

Isaac, N.J.B., Turvey, S.T., Collen, B., Waterman, C. and Baillie, J.E.M. (2007). Mammals on the EDGE: conservation priorities based on threat and phylogeny. *PLoS ONE*, **2**, e296.

See Also

`evol.distinct`, `phyloregion`

Examples

```
tree <- ape::rcoal(10)
evol_distinct(tree)
evol_distinct(tree, type = "fair.proportion")
```

`fishnet` *Create fishnet of regular grids*

Description

The `fishnet` function creates a regular grid of locations covering the study area at various grain sizes.

Usage

```
fishnet(mask, res = 0.5)
```

Arguments

`mask`	a vector polygon covering the boundary of the survey region.
`res`	the grain size of the grid cells in decimal degrees (default).

Value

A spatial vector polygon object of equal area grid cells covering the defined area.

References

Phillips, S.J., Anderson, R.P. & Schapire, R.E. (2006) Maximum entropy modeling of species geographic distributions. *Ecological Modelling* **190**: 231-259.

Examples

```
d <- terra::vect(system.file("ex/nigeria.json", package="phyloregion"))
f <- fishnet(d, res = 0.75)
```

`fitgom` *Fits Grade of membership models for biogeographic regionalization*

Description

Generates grade of membership, “admixture”, “topic” or “Latent Dirichlet Allocation” models, by representing sampling units as partial memberships in multiple groups. It can group regions based on phylogenetic information or functional traits.

Usage

```
fitgom(
  x,
  trait = NULL,
  cut = NULL,
  phy = NULL,
  bin = 10,
  na.rm = FALSE,
  K,
  shape = NULL,
  initopics = NULL,
  tol = 0.1,
  bf = TRUE,
  kill = 2,
  ord = TRUE,
  verb = 1,
  ...
)
```

Arguments

`x`	A community data in long format with one column representing sites labeled “grids” and another column representing species labeled “species”.
`trait`	A data frame or matrix object with the first column labeled “species” containing the taxonomic groups to be evaluated whereas the remaining columns have the various functional traits. The variables must be a mix of numeric and categorical values.
`cut`	The slice time for the phylogenetic tree.
`phy`	is a dated phylogenetic tree with branch lengths stored as a phylo object (as in the ape package).
`bin`	The desired number of clusters or bins.
`na.rm`	Logical, whether NA values should be removed or not.
`K`	The number of latent topics. If length(K)>1, topics will find the Bayes factor (vs a null single topic model) for each element and return parameter estimates for the highest probability K.

`shape`	Optional argument to specify the Dirichlet prior concentration parameter as shape for topic-phrase probabilities. Defaults to 1/(K*ncol(counts)). For fixed single K, this can also be a ncol(counts) by K matrix of unique shapes for each topic element.
`initopics`	Optional start-location for $[\theta_1, \ldots, \theta_K]$, the topic-phrase probabilities. Dimensions must accord with the smallest element of K. If NULL, the initial estimates are built by incrementally adding topics.
`tol`	An indicator for whether or not to calculate the Bayes factor for univariate K. If length(K)>1, this is ignored and Bayes factors are always calculated.
`bf`	An indicator for whether or not to calculate the Bayes factor for univariate K. If length(K)>1, this is ignored and Bayes factors are always calculated.
`kill`	For choosing from multiple K numbers of topics (evaluated in increasing order), the search will stop after kill consecutive drops in the corresponding Bayes factor. Specify kill=0 if you want Bayes factors for all elements of K.
`ord`	If `TRUE`, the returned topics (columns of theta) will be ordered by decreasing usage (i.e., by decreasing `colSums(omega)`).
`verb`	A switch for controlling printed output. verb > 0 will print something, with the level of detail increasing with verb.
`...`	Further arguments passed to or from other methods.

Details

Mapping phylogenetic regions (phyloregions) involves successively slicing the phylogenetic tree at various time depths (e.g., from 1, 2, 3, 4, to 5 million years ago (Ma)), collapsing nodes and ranges that originated at each time depth, and generating a new community matrix based on the presence or absence of each lineage in a grid cell. A grade of membership model is then fitted to the reduced community matrix. To map functional trait regions (traitregions), the function uses k-means to cluster species based on their functional traits, often for mixed-type data including categorical and numeric functional traits. The ranges for each species in each resulting cluster are collapsed to generate a new community matrix based on the presence or absence of cluster representative in a grid cell. A grade of membership model is then fitted to the new reduced community matrix. Mapping bioregions for taxonomic diversity is based on fitting a grade of membership model directly to the original community matrix that is often represented with species in the columns and sites as rows.

Value

An topics object list with entries

- `K` The number of latent topics estimated. If input `length(K)>1`, on output this is a single value corresponding to the model with the highest Bayes factor.
- `theta` The `ncol(counts)` by K matrix of estimated topic-phrase probabilities.
- `omega` The `nrow(counts)` by K matrix of estimated document-topic weights.
- `BF` The log Bayes factor for each number of topics in the input K, against a null single topic model.

- `D` Residual dispersion: for each element of K, estimated dispersion parameter (which should be near one for the multinomial), degrees of freedom, and p-value for a test of whether the true dispersion is >1.
- `X` The input community matrix as a sparse matrix.

Examples

```
library(terra)
data(africa)
names(africa)
p <- vect(system.file("ex/sa.json", package = "phyloregion"))
m <- fitgom(x=sparse2long(africa$comm), K=3)

COLRS <- phyloregion:::hue(m$K)
plot_spatial_membership(m$omega, pol = p, col=COLRS, type="pie")
```

`functional_beta` *Functional beta diversity for mixed-type functional traits*

Description

Computes turnover of functional diversity using k-prototypes clustering algorithm tailored for mixed-type functional traits (numeric and categorical) to generate an integer vector of cluster assignments. The ranges of each species in a cluster are collapsed to generate a new community matrix based on the presence or absence of cluster membership in a grid cell. A grade of membership model or beta diversity is then fitted to the new reduced community matrix for further analysis.

Usage

```
functional_beta(
  x,
  trait = NULL,
  bin = 10,
  na.rm = "no",
  quick_elbow = FALSE,
  abundance = FALSE,
  ...
)
```

Arguments

`x`	A dataframe or sparse community matrix of species occurrences.
`trait`	A data frame with the first column labeled “species” containing the taxonomic groups to be evaluated whereas the remaining columns contain the various functional traits. The variables should be mixed-type combining numeric and categorical variables.

`bin`	The desired number of clusters or bins. If `elbow=TRUE`, the optimal number of clusters is determined by running the analysis multiple times varying from 2 to bin.
`na.rm`	Logical, whether NA values should be removed prior to computation
`quick_elbow`	Quickly estimate the 'elbow' of a scree plot to determine the optimal number of clusters.
`abundance`	Logical, whether the reduced matrix should be returned as presence or absence of cluster representation or as abundances of cluster memberships
`...`	Further arguments passed to or from other methods.

Value

A list with three dissimilarity matrices capturing: (i) turnover (replacement), (ii) nestedness-resultant component, and (iii) total dissimilarity (i.e. the sum of both components).

For index.family="sorensen" the three matrices are:

- `beta.sim` A distance object, dissimilarity matrix accounting for spatial turnover (replacement), measured as Simpson pair-wise dissimilarity.
- `beta.sne` `dist` object, dissimilarity matrix accounting for nestedness-resultant dissimilarity, measured as the nestedness-fraction of Sorensen pair-wise dissimilarity
- `beta.sor` `dist` object, dissimilarity matrix accounting for total dissimilarity, measured as Sorensen pair-wise dissimilarity (a monotonic transformation of beta diversity)

For index.family="jaccard" the three matrices are:

- `beta.jtu` A distance object, dissimilarity matrix accounting for spatial turnover, measured as the turnover-fraction of Jaccard pair-wise dissimilarity
- `beta.jne` `dist` object, dissimilarity matrix accounting for nestedness-resultant dissimilarity, measured as the nestedness-fraction of Jaccard pair-wise dissimilarity
- `beta.jac` `dist` object, dissimilarity matrix accounting for beta diversity, measured as Jaccard pair-wise dissimilarity (a monotonic transformation of beta diversity)

References

Szepannek, G. (2018) clustMixType: User-friendly clustering of mixed-type data in R. *The R Journal*, **10**: 200-208.

Examples

```
library(terra)
data(africa)
p <- vect(system.file("ex/sa.json", package = "phyloregion"))
fb <- functional_beta(x=africa$comm, trait = africa$trait)
p <- phyloregion(fb[[1]], pol = p)
plot(p)
```

`get_clades` *Get descendant nodes of phylogeny at a given time depth*

Description

`get_clades` returns the tips that descend from a given node or time depth on a dated phylogenetic tree.

Usage

```
get_clades(tree, cut = NULL, k = NULL)
```

Arguments

`tree`	is a dated phylogenetic tree with branch lengths stored as a phylo object (as in the ape package).
`cut`	the slice time
`k`	number of slices

Value

A list of descendants

References

Schliep, K.P. (2010) phangorn: phylogenetic analysis in R. *Bioinformatics* **27**: 592–593.

Examples

```
require(ape)
data(bird.orders)
plot(bird.orders)
axisPhylo(side = 1)
abline(v=28-23) # the root is here at 28
get_clades(bird.orders, 23)
```

`hexcols` *Generate diverging colors in HCL colour space.*

Description

A function to generate colors in Hue-Chroma-Luminance colour scheme for mapping phyloregions.

Usage

```
hexcols(x)
```

Arguments

`x`	An object of class `metaMDS`

Value

A range of discrete colors differentiating between phyloregions in terms of their shared relationships.

Author(s)

Barnabas H. Daru `<darunabas@gmail.com>`

Examples

```
library(vegan)
data(dune)
c1 <- metaMDS(dune, trace = 0)
hexcols(c1)
plot(c1$points, pch = 21, cex = 7, bg = hexcols(c1), las = 1)
```

`indicators` *Top driving species in phyloregions*

Description

This function applies a KL-divergence approach to a list of indicator species in phyloregions.

Usage

```
indicators(
  theta,
  top_indicators = 5,
  method = c("poisson", "bernoulli"),
  options = c("min", "max"),
  shared = FALSE
)
```

Arguments

`theta`	A matrix or data.frame of cluster probability distributions from a topics modeling.
`top_indicators`	Integer to obtain the top driving species in clusters.
`method`	The model assumption for KL divergence measurement. Available choices are "poisson" (default) and "bernoulli".
`options`	Option "min" selects species that maximize the minimum KL divergence of a phyloregion vs all other phyloregions. Option "max" selects species that maximize the maximum KL divergence of a phyloregion against all other phyloregions.
`shared`	Logical if TRUE, lists top species driving patterns in more than one phyloregion.

Value

A list of top indicator species and their indicator values

Examples

```
data(africa)
indsp <- indicators(africa$theta, top_indicators = 5,
                    options = "max", method = "poisson")
```

`long2sparse` *Conversion of community data*

Description

These functions convert a community data to compressed sparse matrix, dense matrix and long format (e.g. species records).

Usage

```
long2sparse(x, grids = "grids", species = "species")

sparse2long(x)

dense2sparse(x)

sparse2dense(x)

long2dense(x)

dense2long(x)
```

Arguments

`x`	A community data which one wants to transform
`grids`	column name of the column containing grid cells
`species`	column name of the column containing the species / taxa names

Value

A compressed sparse community matrix of sites by species

Examples

```
data(africa)
africa$comm[1:5, 1:20]
long <- sparse2long(africa$comm)
long[1:5, ]
sparse <- long2sparse(long)
all.equal(africa$comm, sparse)

dense_comm <- matrix(c(1,0,1,1,0,0,
                1,0,0,1,1,0,
                1,1,1,1,1,1,
                0,0,1,1,0,1), 6, 4,
              dimnames=list(paste0("g",1:6), paste0("sp", 1:4)))
dense_comm
sparse_comm <- dense2sparse(dense_comm)
sparse_comm
sparse2long(sparse_comm)
```

`map_trait` *Map species' trait values in geographic space*

Description

`map_trait` add species trait values to species distribution in geographic space.

Usage

```
map_trait(x, trait, FUN = sum, pol = NULL, ...)
```

Arguments

`x`	A community data object - a vector (with names matching trait data) or a data.frame or matrix (with column names matching names in trait data)
`trait`	A data.frame of species traits with a column of species names matching species names in the community data, and another column with the trait values.
`FUN`	The function used to aggregate species trait values in geographic space. By default, if `FUN = sum`, the sum of all species traits per area or grid cell is calculated.
`pol`	a vector polygon of grid cells.
`...`	Further arguments passed to or from other methods.

Value

A data frame of species traits by site.

Author(s)

Barnabas H. Daru `<darunabas@gmail.com>`

Examples

```
data(africa)
library(terra)
p <- vect(system.file("ex/sa.json", package = "phyloregion"))
x <- EDGE(africa$IUCN, africa$phylo, Redlist = "IUCN",
          species = "Species")
y <- map_trait(africa$comm, x, FUN = sd, pol = p)

plot(y, "traits", col = hcl.colors(n=20, palette = "Blue-Red 3", rev=FALSE))
```

`match_phylo_comm` *Match taxa and in phylogeny and community matrix*

Description

match_phylo_comm compares taxa (species, labels, tips) present in a phylogeny with a community matrix. Pruning, sorting and trying to add missing species on genus level if possible to match in subsequent analysis.

Usage

```
match_phylo_comm(phy, comm, delete_empty_rows = TRUE)
```

Arguments

`phy`	A phylogeny
`comm`	A (sparse) community data matrix
`delete_empty_rows`	delete rows with no observation

Details

Based on the function of the same name in picante but allows sparse matrices and with taxa addition.

Value

A list containing the following elements, pruned and sorted to match one another:

`phy`	A phylogeny object of class phylo
`comm`	A (sparse) community data matrix

Examples

```
data(africa)
tree <- africa$phylo
x <- africa$comm

subphy <- match_phylo_comm(tree, x)$phy
submat <- match_phylo_comm(tree, x)$com
```

`mean_dist` *Mean distance matrix from a set of distance matrices*

Description

This function generates the mean pairwise distance matrix from a set many pairwise distance matrices. Note: all matrices should be of the same dimension.

Usage

```
mean_dist(files, trace = 1, ...)
```

Arguments

`files`	list of pairwise distance matrices stored as CSVs or .rds with the same dimensions.
`trace`	Trace the function; trace = 2 or higher will be more voluminous.
`...`	Further arguments passed to or from other methods.

Value

average distance matrix

`nodepie` *Label phylogenetic nodes using pie*

Description

Label phylogenetic nodes using pie

Usage

```
nodepie(
  pie,
  radius = 2,
  pie_control = list(),
  legend = FALSE,
  col = hcl.colors(5),
  ...
)
```

Arguments

`pie`	Estimates from ancestral character reconstruction
`radius`	Radius of the pie
`pie_control`	The list of control parameters to be passed into the add.pie function.
`legend`	Logical, whether to add a legend or not.
`col`	List of colors for the pies.
`...`	Further arguments passed to or from other methods.

Value

Returns no value, just add color pies on phylogenetic nodes!

`optimal_phyloregion` *Determine optimal number of clusters*

Description

This function divides the hierarchical dendrogram into meaningful clusters ("phyloregions"), based on the ‘elbow’ or ‘knee’ of an evaluation graph that corresponds to the point of optimal curvature.

Usage

```
optimal_phyloregion(x, method = "average", k = 20)
```

Arguments

`x`	a numeric matrix, data frame or “dist” object.
`method`	the agglomeration method to be used. This should be (an unambiguous abbreviation of) one of “ward.D”, “ward.D2”, “single”, “complete”, “average” (= UPGMA), “mcquitty” (= WPGMA), “median” (= WPGMC) or “centroid” (= UPGMC).
`k`	numeric, the upper bound of the number of clusters to compute. DEFAULT: 20 or the number of observations (if less than 20).

Value

a list containing the following as returned from the GMD package (Zhao et al. 2011):

- `k`: optimal number of clusters (bioregions)
- `totbss`: total between-cluster sum-of-square
- `tss`: total sum of squares of the data
- `ev`: explained variance given k

References

Salvador, S. & Chan, P. (2004) *Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms*. Proceedings of the Sixteenth IEEE International Conference on Tools with Artificial Intelligence, pp. 576–584. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, USA.

Zhao, X., Valen, E., Parker, B.J. & Sandelin, A. (2011) Systematic clustering of transcription start site landscapes. *PLoS ONE* **6**: e23409.

Examples

```
data(africa)
tree <- africa$phylo
bc <- beta_diss(africa$comm)
(d <- optimal_phyloregion(bc[[1]], k=15))
plot(d$df$k, d$df$ev, ylab = "Explained variances",
  xlab = "Number of clusters")
lines(d$df$k[order(d$df$k)], d$df$ev[order(d$df$k)], pch = 1)
points(d$optimal$k, d$optimal$ev, pch = 21, bg = "red", cex = 3)
points(d$optimal$k, d$optimal$ev, pch = 21, bg = "red", type = "h")
```

`PD` *Phylogenetic diversity*

Description

`PD` calculates Faith's (1992) phylogenetic diversity and relative phylogenetic diversity.

Usage

```
PD(x, phy)

RPD(x, phy)
```

Arguments

`x`	a community matrix, i.e. an object of class matrix or Matrix or an object of class phyloseq.
`phy`	a phylogenetic tree (object of class phylo).

Value

a vector with the PD for all samples.

References

Faith, D.P. (1992) Conservation evaluation and phylogenetic diversity. *Biological Conservation* **61**: 1–10.

See Also

read.community read.tree phylobeta_core

Examples

```
library(ape)
library(Matrix)
tree <- read.tree(text ="((t1:1,t2:1)N2:1,(t3:1,t4:1)N3:1)N1;")
com <- sparseMatrix(c(1,3,4,1,4,5,1,2,3,4,5,6,3,4,6),
  c(1,1,1,2,2,2,3,3,3,3,3,3,4,4,4),x=1,
  dimnames = list(paste0("g", 1:6), tree$tip.label))

PD(com, tree)
# Relative phylogenetic diversity
RPD(com, tree)
```

PD_ses *Phylogenetic diversity standardized for species richness*

Description

This function computes the standard effect size of PD by correcting for changes in species richness. The novelty of this function is its ability to utilize sparse community matrix making it possible to efficiently randomize very large community matrices spanning thousands of taxa and sites.

Usage

```
PD_ses(
  x,
  phy,
  model = c("tipshuffle", "rowwise", "colwise"),
  reps = 10,
  metric = "pd",
  ...
)
```

Arguments

`x`	a (sparse) community matrix, i.e. an object of class matrix or Matrix.
`phy`	a phylogenetic tree (object of class phylo).
`model`	The null model for separating patterns from processes and for contrasting against alternative hypotheses. Available null models include: • “tipshuffle”: shuffles tip labels multiple times. • “rowwise”: shuffles sites (i.e., varying richness) and keeping species occurrence frequency constant. • “colwise”: shuffles species occurrence frequency and keeping site richness constant.

`reps`	Number of replications.
`metric`	The phylodiversity measure to compute.
`...`	Further arguments passed to or from other methods.

Value

A data frame of results for each community or grid cell

- grids: Site identity
- richness: Number of taxa in community
- pd_obs: Observed PD in community
- pd_rand.mean: Mean PD in null communities
- pd_rand.sd: Standard deviation of PD in null communities
- pd_obs.rank: Rank of observed PD vs. null communities
- pd_obs.z: Standardized effect size of PD vs. null communities $= (pd_obs-pd_rand.mean)/pd_rand_sd$
- pvalue: P-value (quantile) of observed PD vs. null communities $= mpd_obs_rank/iter+1$
- reps: Number of replicates
- p_obs_c_lower: Number of times observed value < random value
- p_obs_c_upper: Number of times observed value > random value
- p_obs_p_lower: Percentage of times observed value < random value
- p_obs_p_upper: Percentage of times observed value > random value
- p_obs_q: Number of the non-NA random values used for comparison

References

Proches, S., Wilson, J.R.U. & Cowling, R.M. (2006) How much evolutionary history in a 10 x 10m plot? *Proceedings of Royal Society B* **273**: 1143-1148.

Examples

```
library(ape)
library(Matrix)
tree <- read.tree(text ="((t1:1,t2:1)N2:1,(t3:1,t4:1)N3:1)N1;")
com <- sparseMatrix(c(1,3,4,1,4,5,1,2,3,4,5,6,3,4,6),
  c(1,1,1,2,2,2,3,3,3,3,3,3,4,4,4),x=1,
  dimnames = list(paste0("g", 1:6), tree$tip.label))

PD_ses(com, tree, model="rowwise")
```

`phylobeta_core` *Phylogenetic beta diversity*

Description

`phylobeta_core` computes efficiently for large community matrices and trees the necessary quantities used by the betapart package to compute pairwise and multiple-site phylogenetic dissimilarities.

Usage

```
phylobeta_core(x, phy)

phylobeta(x, phy, index.family = "sorensen")
```

Arguments

`x`	an object of class Matrix, matrix or phyloseq
`phy`	a phylogenetic tree (object of class phylo)
`index.family`	family of dissimilarity indices, partial match of "sorensen" or "jaccard".

Value

`phylobeta_core` returns an object of class "phylo.betapart", see `phylo.betapart.core` for details. This object can be called by `phylo.beta.pair` or `phylo.beta.multi`.

`phylobeta` returns a list with three phylogenetic dissimilarity matrices. See `phylo.beta.pair` for details.

Author(s)

Klaus Schliep

See Also

`read.community`, `phylo.betapart.core`, `beta_core`

Examples

```
library(ape)
tree <- read.tree(text = "((t1:1,t2:1)N2:1,(t3:1,t4:1)N3:1)N1;")
com <- sparseMatrix(c(1,3,4,1,4,5,1,2,3,4,5,6,3,4,6),
  c(1,1,1,2,2,2,3,3,3,3,3,3,4,4,4),x=1,
  dimnames = list(paste0("g", 1:6), tree$tip.label))
com

pbc <- phylobeta_core(com, tree)
pb <- phylobeta(com, tree)
```

`phylobeta_ses` *Phylogenetic beta diversity standardized for species beta diversity*

Description

This function computes the standard effect size of phylogenetic beta diversity by correcting for changes in species beta diversity. The novelty of this function is its ability to utilize sparse community matrix making it possible to efficiently randomize very large community matrices spanning thousands of taxa and sites.

Usage

```
phylobeta_ses(
  x,
  phy,
  index.family = "simpson",
  model = c("tipshuffle", "rowwise", "colwise"),
  reps = 1000,
  ...
)
```

Arguments

`x`	a (sparse) community matrix, i.e., an object of class matrix or Matrix.
`phy`	a phylogenetic tree (object of class phylo).
`index.family`	the family of dissimilarity indices including “simpson”, “sorensen” and “jaccard”.
`model`	The null model for separating patterns from processes and for contrasting against alternative hypotheses. Available null models include: • “tipshuffle”: shuffles phylogenetic tip labels multiple times. • “rowwise”: shuffles sites (i.e., varying richness) and keeping species occurrence frequency constant. • “colwise”: shuffles species occurrence frequency and keeping site richness constant.
`reps`	Number of replications.
`...`	Further arguments passed to or from other methods.

Value

A data frame of results for each community or grid cell

- phylobeta_obs: Observed phylobeta in community
- phylobeta_rand_mean: Mean phylobeta in null communities
- phylobeta_rand_sd: Standard deviation of phylobeta in null communities

- phylobeta_obs_z: Standardized effect size of phylobeta vs. null communities $= (phylobeta_obs - phylobeta_rand_mean)/phylobeta_rand_sd$
- reps: Number of replicates

References

Proches, S., Wilson, J.R.U. & Cowling, R.M. (2006) How much evolutionary history in a 10 x 10m plot? *Proceedings of Royal Society B* **273**: 1143-1148.

Examples

```
library(ape)
library(Matrix)
tree <- read.tree(text ="((t1:1,t2:1)N2:1,(t3:1,t4:1)N3:1)N1;")
com <- sparseMatrix(c(1,3,4,1,4,5,1,2,3,4,5,6,3,4,6),
  c(1,1,1,2,2,2,3,3,3,3,3,3,4,4,4),x=1,
  dimnames = list(paste0("g", 1:6), tree$tip.label))

phylobeta_ses(com, tree, model="rowwise")
```

`phylobuilder` *Create a subtree with largest overlap from a species list.*

Description

phylobuilder creates a subtree with largest overlap from a species list. If species in the species list are not already in the tip label, species will be added at the most recent common ancestor at the genus or family level when possible.

Usage

```
phylobuilder(species, tree, extract = TRUE)
```

Arguments

`species`	A vector or matrix containing a species list
`tree`	a phylogenetic tree (object of class phylo)
`extract`	extract the species in the list after trying to add missing labels to the tree. If FALSE phylobuilder adds only the taxa in the list.

Value

`phylobuilder` returns a phylogenetic tree, i.e. an object of class `phylo`.

See Also

`add.tips`, `label2table`, `stripLabel`

Examples

```
library(ape)
txt <- "(((((Panthera_leo,Panthera_pardus), Panthera_onca),(Panthera_uncia,
  (Panthera_tigris_altaica, Panthera_tigris_amoyensis)))Panthera)Felidae,
  (((((((Canis_lupus,Canis_lupus_familiaris),Canis_latrans),Canis_anthus),
  Canis_aureus),Lycaon_pictus),(Canis_adustus,Canis_mesomelas))Canis)
  Canidae)Carnivora;"
txt <- gsub("[[:space:]]", "", txt)
cats_and_dogs <- read.tree(text=txt)
plot(cats_and_dogs, node.depth=2, direction="downwards")
nodelabels(cats_and_dogs$node.label, frame="none", adj = c(0.5, 0))

tree <- drop.tip(cats_and_dogs, c("Panthera_uncia", "Lycaon_pictus"),
  collapse.singles=FALSE)

dogs <- c("Canis_lupus", "Canis_lupus_familiaris", "Canis_latrans",
  "Canis_anthus", "Canis_aureus", "Lycaon_pictus", "Canis_adustus",
  "Canis_mesomelas")

# try to extract tree with all 'dogs'
t1 <- phylobuilder(dogs, tree)
plot(t1, direction="downwards")
attr(t1, "species_list")

# providing extra information ("Family", "Order", ...) can help
sp <- data.frame(Order = c("Carnivora", "Carnivora", "Carnivora"),
  Family = c("Felidae", "Canidae", "Canidae"),
  Genus = c("Panthera", "Lycaon", "Vulpes"),
  Species = c("uncia", "pictus", "vulpes"),
  Common_name = c("Snow leopard", "Africa wild dog", "Red fox"))
sp
# Now we just add some species
t2 <- phylobuilder(sp, tree, extract=FALSE)
plot(t2, direction="downwards")
attr(t2, "species_list")
```

`phyloregion` *Compute phylogenetic regionalization and evolutionary distinctiveness of phyloregions*

Description

This function estimates evolutionary distinctiveness of each phyloregion by computing the mean value of phylogenetic beta diversity between a focal phyloregion and all other phyloregions in the study area.

Usage

```
phyloregion(x, k = 10, method = "average", pol = NULL, ...)

infomap(x, pol = NULL, ...)
```

Arguments

`x`	A distance matrix
`k`	The desired number of phyloregions, often as determined by `optimal_phyloregion`.
`method`	the agglomeration method to be used. This should be (an unambiguous abbreviation of) one of “ward.D”, “ward.D2”, “single”, “complete”, “average” (= UPGMA), “mcquitty” (= WPGMA), “median” (= WPGMC) or “centroid” (= UPGMC).
`pol`	a vector polygon of grid cells or spatial points.
`...`	Further arguments passed to or from other methods.

Value

An object of class `phyloregion` containing

- a data frame membership with columns grids and cluster
- k the number of clusters and additionally there can be an shape file and other objects. This representation may still change.

Author(s)

Barnabas H. Daru <`darunabas@gmail.com`>

References

Daru, B.H., Van der Bank, M., Maurin, O., Yessoufou, K., Schaefer, H., Slingsby, J.A. & Davies, T.J. (2016) A novel phylogenetic regionalization of the phytogeographic zones of southern Africa reveals their hidden evolutionary affinities. *Journal of Biogeography* **43**: 155-166.

Daru, B.H., Elliott, T.L., Park, D.S. & Davies, T.J. (2017) Understanding the processes underpinning patterns of phylogenetic regionalization. *Trends in Ecology and Evolution* **32**: 845-860.

Daru, B.H., Holt, B.G., Lessard, J.P., Yessoufou, K. & Davies, T.J. (2017) Phylogenetic regionalization of marine plants reveals close evolutionary affinities among disjunct temperate assemblages. *Biological Conservation* **213**: 351-356.

See Also

`evol_distinct`, `optimal_phyloregion`, `evol.distinct` for a different approach.

Examples

```
library(ape)
tree <- read.tree(text = "((t1:1,t2:1)N2:1,(t3:1,t4:1)N3:1)N1;")
com <- sparseMatrix(c(1,3,4,1,4,5,1,2,3,4,5,6,3,4,6),
  c(1,1,1,2,2,2,3,3,3,3,3,3,4,4,4),x=1,
  dimnames = list(paste0("g", 1:6), tree$tip.label))
pbc <- phylobeta(com, tree)
# phyloregion(pbc[[1]], k = 3)
```

`phylo_endemism` *Phylogenetic Endemism*

Description

Calculates phylogenetic endemism (sum of 'unique' branch lengths) of multiple ecological samples.

Usage

```
phylo_endemism(x, phy, weighted = TRUE)
```

Arguments

`x`	is the community data given as a data.frame or matrix with species/OTUs as columns and samples/sites as rows (like in the vegan package). Columns are labeled with the names of the species/OTUs. Rows are labelled with the names of the samples/sites. Data can be either abundance or incidence (0/1). Column labels must match tip labels in the phylogenetic tree exactly!
`phy`	a (rooted) phylogenetic tree (phylo) with branch lengths
`weighted`	is a logical indicating whether weighted endemism (default) or strict endemism should be calculated.

Details

Takes a community data table and a (rooted) phylogenetic tree (with branch lengths) and calculates either strict or weighted endemism in Phylogenetic Diversity (PD). Strict endemism equates to the total amount of branch length found only in the sample/s and is described by Faith et al. (2004) as PD-endemism. Weighted endemism calculates the "spatial uniqueness" of each branch in the tree by taking the reciprocal of its range, multiplying by branch length and summing for all branch lengths present at a sample/site. Range is calculated simply as the total number of samples/sites at which the branch is present. This latter approach is described by Rosauer et al. (2009) as Phylogenetic endemism.

Value

`phylo_endemism` returns a vector of phylogenetic endemism for each sample or site.

References

Faith, D.P., Reid, C.A.M. & Hunter, J. (2004) Integrating phylogenetic diversity, complementarity, and endemism for conservation assessment. *Conservation Biology* **18**(1): 255-261.

Rosauer, D., Laffan, S.W., Crisp, M.D., Donnellan, C. & Cook, L.G. (2009). Phylogenetic endemism: a new approach for identifying geographical concentrations of evolutionary history. *Molecular Ecology* **18**(19): 4061-4072.

Daru, B.H., Farooq, H., Antonelli, A. & Faurby, S. (2020) Endemism patterns are scale dependent. *Nature Communications* **11** : 2115.

Examples

```
data(africa)
pe <- phylo_endemism(africa$comm, africa$phylo)
plot(density(pe))
```

`plot.phyloregion` *Visualize biogeographic patterns*

Description

Visualize biogeographic patterns

Usage

```
## S3 method for class 'phyloregion'
plot(x, pol = NULL, palette = "NMDS", col = NULL, label = FALSE, ...)

plot_NMDS(x, ...)

text_NMDS(x, ...)
```

Arguments

`x`	an object of class phyloregion from `phyloregion`
`pol`	a polygon shapefile of grid cells.
`palette`	name of the palette to generate colors from. The default, “NMDS”, allows display of phyloregions in multidimensional scaling color space matching the color vision of the human visual system. The name is matched to the list of available color palettes from the `hcl.colors` function in the `grDevices` package.
`col`	vector of colors of length equal to the number of phyloregions.
`label`	Logical, whether to print cluster names or not
`...`	arguments passed among methods.

Value

No return value, called for plotting.

Examples

```
library(terra)
data(africa)
tree <- africa$phylo
x <- africa$comm
p <- vect(system.file("ex/sa.json", package = "phyloregion"))

subphy <- match_phylo_comm(tree, x)$phy
submat <- match_phylo_comm(tree, x)$com

pbc <- phylobeta(submat, subphy)
y <- phyloregion(pbc[[1]], pol=p)

plot_NMDS(y, cex=6)
text_NMDS(y, cex=2)
plot(y, cex=1, palette="NMDS")
plot(y, cex=1)
```

`plot.sparse` *Create illustrative sparse matrix*

Description

This function visualizes a sparse matrix using vertical bands corresponding to presence or absence of a species in an area.

Usage

```
## S3 method for class 'sparse'
plot(x, col = c("red", "yellow"), lwd = 1, ...)
```

Arguments

`x`	A matrix
`col`	A vector of colors to represent presence or absence of a species
`lwd`	Line width
`...`	Further arguments passed to or from other methods.

Value

Returns no value, just plot sparse matrix

`plot_spatial_membership` *Visualize spatial memberships*

Description

Visualize spatial memberships

Usage

```
plot_spatial_membership(
  omega,
  pol,
  radius = NULL,
  col = hcl.colors(5),
  pie_control = list(),
  legend = FALSE,
  legend_pie = FALSE,
  type = c("pie", "blend"),
  ...
)
```

Arguments

`omega`	a matrix of phyloregion of probabilities of each species
`pol`	a vector polygon of grid cells with a column labeled "grids".
`radius`	Radius of the pie legend to be displayed
`col`	List of colors for the pies.
`pie_control`	The list of control parameters to be passed into the add.pie function.
`legend`	Logical, whether to plot a legend or not.
`legend_pie`	Legend for the pie plots.
`type`	Type of visualization, whether piecharts or blended colors.
`...`	Further arguments passed to or from other methods.

Value

Returns no value, just map color pies in geographic space!

Examples

```
library(terra)
data(africa)
p <- vect(system.file("ex/sa.json", package = "phyloregion"))
K <- ncol(africa$omega)

CLRS <- hcl.colors(K)
plot_spatial_membership(africa$omega, pol = p, col=CLRS, type="blend")
```

`random_species` *Generate random species distributions in space*

Description

This function generates random species distributions in geographic space as extent of occurrence range polygons based on convex hulls of random points.

Usage

```
random_species(n, species, pol, ...)
```

Arguments

`n`	vector of one or more elements to choose from, or a positive integer.
`species`	the desired number of species.
`pol`	the vector polygon of the study area for determining the species distributions
`...`	Further arguments passed to or from other methods.

Value

A vector polygon of species' extent of occurrence ranges.

Author(s)

Barnabas H. Daru `<darunabas@gmail.com>`

`rast2comm` *Convert raw input distribution data to community*

Description

The functions `points2comm`, `polys2comm`, `rast2comm` provide convenient interfaces to convert raw distribution data often available as point records, polygons and raster layers, respectively, to a community composition data frame at varying spatial grains and extents for downstream analyses.

Usage

```
rast2comm(files)

polys2comm(dat, res = 0.25, pol.grids = NULL, ...)

points2comm(dat, res = 0.25, pol.grids = NULL, ...)
```

Arguments

`files`	list of SpatRaster layer objects with the same spatial extent and resolution.
`dat`	layers of merged maps corresponding to species polygons for `polys2comm`; or point occurrence data frame for `points2comm`, with at least three columns: • Column 1: `species` (listing the taxon names) • Column 2: `decimallongitude` (corresponding to decimal longitude) • Column 3: `decimallatitude` (corresponding to decimal latitude)
`res`	the grain size of the grid cells in decimal degrees (default).
`pol.grids`	if specified, the vector polygon of grid cells with a column labeled “grids”.
`...`	Further arguments passed to or from other methods.

Value

Each of these functions generate a list of two objects as follows:

- comm_dat: (sparse) community matrix
- map: vector or raster of grid cells with the values per cell for mapping.

See Also

`mapproject` for conversion of latitude and longitude into projected coordinates system. `long2sparse` for conversion of community data.

Examples

```
fdir <- system.file("NGAplants", package="phyloregion")
files <- file.path(fdir, dir(fdir))
ras <- rast2comm(files) # Note, this function generates
     # a list of two objects
head(ras[[1]])



require(terra)
s <- vect(system.file("ex/nigeria.json", package="phyloregion"))
sp <- random_species(100, species=5, pol=s)
pol <- polys2comm(dat = sp)
head(pol[[1]])


library(terra)
s <- vect(system.file("ex/nigeria.json", package="phyloregion"))
set.seed(1)
m <- as.data.frame(spatSample(s, 1000, method = "random"),
                   geom = "XY")[-1]
names(m) <- c("lon", "lat")
species <- paste0("sp", sample(1:100))
m$taxon <- sample(species, size = nrow(m), replace = TRUE)
```

```
pt <- points2comm(dat = m, res = 0.5) # This generates a list of two objects
head(pt[[1]])
```

`rast_quantile` *Standardizes raster values for mapping*

Description

This function standardizes values of a raster layer for mapping.

Usage

```
rast_quantile(ras, ref)
```

Arguments

`ras`	an input raster layer.
`ref`	a raster layer for reference.

Value

A raster layer that has been standardized and ready for mapping

---

`read.community` *Read in sparse community matrices*

Description

`read.community` reads in file containing occurrence data and returns a sparse matrix.

Usage

```
read.community(file, grids = "grids", species = "species", ...)
```

Arguments

`file`	A file name.
`grids`	Column name of the column containing grid cells.
`species`	Column name of the column containing the species / taxa names.
`...`	further arguments passed to or from other methods.

Value

`read.community` returns a sparse matrix (an object of class "dgCMatrix").

Examples

```
df <- data.frame(grids=paste0("g", c(1,1,2,3,3)),
                 species = paste0("sp", c(1,3,2,1,4)))
df
tmp <- tempfile()
write.csv(df, tmp)
(M <- read.community(tmp) )
sparse2long(M)
unlink(tmp)
```

`sdm` *Fast species distribution model*

Description

This function computes species distribution models using two modelling algorithms: generalized linear models, and maximum entropy (only if `rJava` is available). Note: this is an experimental function, and may change in the future.

Usage

```
sdm(
  x,
  layers = NULL,
  pol = NULL,
  thin = TRUE,
  thin.size = 500,
  algorithm = "all",
  size = 50,
  width = 50000,
  mask = FALSE,
  predictors,
  background = NULL
)
```

Arguments

`x`	A dataframe containing the species occurrences and geographic coordinates. Column 1 labeled as "species", column 2 "lon", column 3 "lat".
`layers`	A `SpatRaster` of predictor variables for fitting species distribution models from species occurrences.
`pol`	A vector polygon specifying the calibration area or boundary to account for a more realistic dispersal capacity and ecological limitation of a species. If `NULL`, the extent of input points is used.
`thin`	Whether to spatially thin occurrences

`thin.size`	The size of the thin occurrences.
`algorithm`	Character. The choice of algorithm to run the species distribution model. For now, the available algorithms include: • "all": Calls all available algorithms: both GLM and MAXENT. • "GLM": Calls only Generalized linear model. • "MAXENT": Calls only Maximum entropy.
`size`	Minimum number of points required to successfully run a species distribution model especially for species with few occurrences.
`width`	Width of buffer in meter if `x` is in longitude/latitude CRS.
`mask`	logical. Should `layers` be used to mask? Only used if `pol` is a SpatVector.
`predictors`	If predicting to new time points, the climate layers for the time points.
`background`	A dataframe of background points, specifying 2 columns with long lat or x and y as nulls for species distribution modeling, often using a vector of probability weights.

Value

A list with the following objects:

- `ensemble_raster` The ensembled raster that predicts the potential species distribution based on the algorithms selected.
- `data` The dataframe of occurrences used to implement the model.
- `polygon` Map polygons of the predicted distributions analogous to extent-of-occurrence range polygon.
- `indiv_models` Raster layers for the separate models that predict the potential species distribution.

References

Zurell, D., Franklin, J., König, C., Bouchet, P.J., Dormann, C.F., Elith, J., Fandos, G., Feng, X., Guillera-Arroita, G., Guisan, A., Lahoz-Monfort, J.J., Leitão, P.J., Park, D.S., Peterson, A.T., Rapacciuolo, G., Schmatz, D.R., Schröder, B., Serra-Diaz, J.M., Thuiller, W., Yates, K.L., Zimmermann, N.E. and Merow, C. (2020), A standard protocol for reporting species distribution models. *Ecography*, **43**: 1261-1277.

Examples

```
# get predictor variables
library(predicts)
f <- system.file("ex/bio.tif", package="predicts")
preds <- rast(f)
#plot(preds)

# get species occurrences
b <- file.path(system.file(package="predicts"), "ex/bradypus.csv")
d <- read.csv(b)
```

```
# fit ensemble model for four algorithms
# m <- sdm(d, layers = preds, predictors = preds, algorithm = "all")
# plot(m$ensemble_raster)
# plot(m$polygon, add=TRUE)
```

`selectbylocation` *Select polygon features from another layer and adds polygon attributes to layer*

Description

The `selectbylocation` function selects features based on their location relative to features in another layer.

Usage

```
selectbylocation(x, y)
```

Arguments

`x`	source layer of the class SpatVect
`y`	Target layer or mask extent to subset from.

Value

A spatial polygons or spatial points object pruned to the extent of the target layer.

Examples

```
library(terra)
d <- vect(system.file("ex/nigeria.json", package="phyloregion"))
e <- ext(d)

set.seed(1)
m <- data.frame(lon = runif(1000, e[1], e[2]),
                lat = runif(1000, e[3], e[4]),
                sites = seq(1000))
m <- vect(m)
z <- selectbylocation(m, d)
plot(d)
points(m, col = "blue", pch = "+")
points(z, col = "red", pch = "+")
```

`select_linkage` *Cluster algorithm selection and validation*

Description

This function contrasts different hierarchical clustering algorithms on the phylogenetic beta diversity matrix for degree of data distortion using Sokal & Rohlf's (1962) cophenetic correlation coefficient.

Usage

```
select_linkage(x)
```

Arguments

`x`	a numeric matrix, data frame or "dist" object.

Value

- A numeric value corresponding to the good clustering algorithm for the distance matrix
- If plot = TRUE, a barplot of cophenetic correlation for all the clustering algorithms is drawn.

References

Sokal, R.R. & Rohlf, F.J. (1962) The comparison of dendrograms by objective methods. *Taxon* **11**: 33–40.

Examples

```
data(africa)
tree <- africa$phylo
bc <- beta_diss(africa$comm)
y <- select_linkage(bc[[1]])
barplot(y, horiz = TRUE, las = 1)
```

`timeslice` *Slice phylogenetic tree at various time depths*

Description

This function slices a dated phylogenetic tree at successive time depths back in time by collapsing younger phylogenetic branches into older ones to infer the origins of species assemblages.

Usage

```
timeslice(phy, n = 0.2, collapse = FALSE, ...)
```

Arguments

`phy`	A dated phylogenetic tree as an object of class “phylo”.
`n`	Time depth to slice the phylogenetic tree (often in millions of years for dated trees).
`collapse`	Logical, collapse internal edges with zero edge length.
`...`	arguments passed among methods.

Value

A tree with the phylogenetic structure removed at the specified time depth

Author(s)

Barnabas H. Daru <`darunabas@gmail.com`>

References

Daru, B.H., van der Bank, M. & Davies, T.J. (2018) Unravelling the evolutionary origins of biogeographic assemblages. *Diversity and Distributions* **24**: 313–324.

Examples

```
library(ape)

set.seed(1)
tree <- rcoal(50)
x <- timeslice(tree, .5)

old.par <- par(no.readonly = TRUE)
par(mfrow = c(1, 2))
plot(tree)
axisPhylo()
plot(x)
axisPhylo()
par(old.par)
```

`tree_sampler` *Subset trees from posterior distribution of trees.*

Description

This function randomly samples a subset of trees from a posterior distribution of trees derived from multiple runs of MrBayes.

Usage

```
tree_sampler(wd, n = 100, pattern = ".tre", ...)
```

Arguments

`wd`	A path to the working directory with the distributions of multiple phylogenetic trees.
`n`	The desired number of subsets of trees. This defaults to 100.
`pattern`	An optional regular expression specifying the file extension.
`...`	arguments passed among methods.

Value

An object of class "multiPhylo" with the subset of trees.

Author(s)

Dominic Bennett & Harith Farooq `<harithmorgadinho@gmail.com>`

`unifrac` *UniFrac distance*

Description

`unifrac` calculates the unweighted UniFrac distance between communities.

Usage

```
unifrac(x, phy)
```

Arguments

`x`	a community matrix, i.e. an object of class matrix or Matrix, or an object of class phyloseq.
`phy`	a phylogenetic tree (object of class phylo).

Value

a dist object.

References

Lozupone C, Knight R. (2005) UniFrac: a new phylogenetic method for comparing microbial communities. *Appl Environ Microbiol.* **71 (12)**:8228–35. *BMC Bioinformatics* 7:371.

See Also

`PD`, `phylobeta`

Examples

```
tree <- ape::read.tree(text ="((t1:1,t2:1)N2:1,(t3:1,t4:1)N3:1)N1;")
com <- Matrix::sparseMatrix(c(1,3,4,1,4,5,1,2,3,4,5,6,3,4,6),
  c(1,1,1,2,2,2,3,3,3,3,3,3,4,4,4),x=1,
  dimnames = list(paste0("g", 1:6), tree$tip.label))

unifrac(com, tree)
```

`weighted_endemism` *Measure the distribution of narrow-ranged or endemic species.*

Description

`weighted_endemism` is species richness inversely weighted by species ranges.

Usage

```
weighted_endemism(x)
```

Arguments

`x`	A (sparse) community matrix.

Value

A data frame of species traits by site.

References

Crisp, M.D., Laffan, S., Linder, H.P. & Monro, A. (2001) Endemism in the Australian flora. *Journal of Biogeography* **28**: 183–198.

Daru, B.H., Farooq, H., Antonelli, A. & Faurby, S. (2020) Endemism patterns are scale dependent. *Nature Communications* **11** : 2115.

Examples

```
library(terra)
data(africa)
p <- vect(system.file("ex/sa.json", package = "phyloregion"))
Endm <- weighted_endemism(africa$comm)
m <- merge(p, data.frame(grids=names(Endm), WE=Endm), by="grids")
m <- m[!is.na(m$WE),]

plot(m, "WE", col = hcl.colors(20), type="continuous")
```

Index

∗ **Conversion**
    long2sparse, 21
∗ **bioregion**
    choropleth, 7
    counts, 10
    fishnet, 14
    fitgom, 15
    get_clades, 19
    indicators, 20
    match_phylo_comm, 23
    phylobuilder, 31
    select_linkage, 44
    selectbylocation, 43
    weighted_endemism, 47
∗ **cluster**
    PD, 26
    read.community, 40
    unifrac, 46
∗ **datasets**
    africa, 3
∗ **phyloregion**
    coldspots, 8
    hexcols, 19
    map_trait, 22
    optimal_phyloregion, 25
    phylobeta_core, 29
    phyloregion, 32
    random_species, 38
    timeslice, 44

add.tips, *31*
africa, 3
arc_labels, 4

backg, 6
beta.multi, *7*
beta.pair, *7*
beta_core, 6, *29*
beta_diss (beta_core), 6
betapart, *7*
betapart.core, *7*

choropleth, 7
coldspots, *8*, 8
collapse_range, 9
counts, 10

dense2long (long2sparse), 21
dense2sparse (long2sparse), 21
dirpath, 11

EDGE, *3*, 12
evol.distinct, *14*, *33*
evol_distinct, 13, *33*

fishnet, *3*, 14
fitgom, 15
functional_beta, 17

get_clades, 19

hexcols, 19
hotspots (coldspots), 8

indicators, 20
infomap (phyloregion), 32

label2table, *31*
long2dense (long2sparse), 21
long2sparse, 21, *39*

map_trait, 22
mapproject, *39*
match_phylo_comm, 23
mean_dist, 24
metaMDS, *20*

nodepie, 24

optimal_phyloregion, 25, *33*

PD, 26, *46*

PD_ses, 27
phylo.beta.multi, *29*
phylo.beta.pair, *29*
phylo.betapart.core, *29*
phylo_endemism, 34
phylobeta, *7*, *46*
phylobeta *(*phylobeta_core*)*, 29
phylobeta_core, 29
phylobeta_ses, 30
phylobuilder, 31
phyloregion, *14*, 32
plot.phyloregion, 35
plot.sparse, 36
plot_NMDS *(*plot.phyloregion*)*, 35
plot_spatial_membership, 37
points2comm *(*rast2comm*)*, 38
polys2comm *(*rast2comm*)*, 38

random_species, 38
randpoints *(*backg*)*, 6
rast2comm, 38
rast_hotspot *(*coldspots*)*, 8
rast_quantile, 40
read.community, *29*, 40
RPD *(*PD*)*, 26

sdm, 41
select_linkage, 44
selectbylocation, 43
sparse2dense *(*long2sparse*)*, 21
sparse2long *(*long2sparse*)*, 21
stripLabel, *31*

text_NMDS *(*plot.phyloregion*)*, 35
timeslice, 44
tree_sampler, 45

unifrac, 46

weighted_endemism, 47