Package ‘specificity’

December 9, 2020

Title Calculate Environmental or Host Phylogenetic Specificity

Version 0.1.0.9000

Description The purpose of this package is to calculate phylogenetic and environmental specificity of species. I wrote this software to analyze specificity of microbes to hosts or to environment, but there is no reason that this software wouldn't work with macroorganisms as well.

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports ape, parallel, Rcpp, fields

LinkingTo Rcpp

NeedsCompilation yes

Author John L Darcy [aut, cre]

Maintainer John L Darcy <darcyj@colorado.edu>

Depends R (>= 3.5.0)

R topics documented:

`bl_distance_ns` *bl_distance_ns*

Description

Calculates branch-length distance between tipa and tipb in a phylogenetic tree using nested-set optomization. Requires a pre-calculated nested-set.

Usage

```
bl_distance_ns(tipa, tipb, tree, ns)
```

Arguments

`tipa`	string. Name of a tip in tree.
`tipb`	string. Name of another tip in tree.
`tree`	phylo object. Tree containing all unique species in x as tips. May contain tips that are not in x.
`ns`	matrix. Nested-set matrix for tree; use make_nested_set(tree).

Value

Distance between tipa and tipb.

Author(s)

John L. Darcy

Examples

```
library(ape)
example_tree <- ape::read.tree(text=" ((((a:1,b:1):1,c:2):1,d:3):1,(e:1,f:1):3);")
plot(example_tree); axis(side=1)
example_ns <- make_nested_set(example_tree)
bl_distance_ns("a", "c", example_tree, example_ns) # should be 4
bl_distance_ns("a", "f", example_tree, example_ns) # should be 8
bl_distance_ns("d", "c", example_tree, example_ns) # should be 6
```

`calculate_spec_and_pval` *calculate_spec_and_pval*

Description

This function is called by phy_or_env_spec(). It is made available as a standalone function in the (rare) case a user wishes to calculate specificity using their own null model. calculate_spec_and_pval() takes empirical rao values and sim rao values (from a null model) and calculates specificity and P-values. To do that, use your own null model to make species data, and use rao1sp() and/or raoperms() to get raw rao values. This function expects a vector of empirical values, and a list of vectors of sim values (see below). Most of the inputs for this function are the same as phy_or_env_spec(). Think of this function as the final component of "build your own phy_or_env_spec()". Note that for this custom approach, the environmental variable must be a dist.

Usage

```
calculate_spec_and_pval(
  emp_raos,
  sim_raos,
  abunds_mat,
  env,
  p_adj = "fdr",
  tails = 1,
  n_cores = 2,
  verbose = TRUE,
  p_method = "raw",
  center = "mean",
  denom_type = "index",
  diagnostic = FALSE
)
```

Arguments

emp_raos	vector. Empirical rao values, one per species ("feature").

`sim_raos`	list of numeric vectors. Sim rao values, generated under null hypothesis. Each item in list corresponds to an entry in emp_raos. As such, length(emp_raos) must equal length(sim_raos). Each item within sim_raos is a vector or rao values (length=n_sim in the case of phy_or_env_spec()).
`abunds_mat`	site x species matrix. See ?phy_or_env_spec.
`env`	MUST BE A dist OBJECT!!!! VERY IMPORTANT!!!! See ?phy_or_env_spec.
`p_adj`	string. Type of multiple hypothesis testing correction performed on P-values. Can take any valid method argument to p.adjust, including "none", "bonferroni", "holm", "fdr", and others (default: "fdr").
`tails`	integer. 1 = 1-tailed, test for specificity only. 2 = 2-tailed. 3 = 1-tailed, test for cosmopolitanism only. 0 = no test, P=1.0 (default: 1).
`n_cores`	integer. Number of CPU cores to use for parallel operations. If set to 1, lapply will be used instead of mclapply (default: 2).
`verbose`	logical. Should status messages be displayed? (default: TRUE).
`p_method`	string. "raw" for quantile method, or "gamma_fit" for calculating P by fitting a gamma distribution (default: "raw").
`center`	string. Type of central tendency to use for simulated RQE values. Options are "mean", "median", and "mode". If mode is chosen, a reversible gamma distribution is fit and mode is calculated using that distribution (default: mean).
`denom_type`	string. Type of denominator (d) to use (default: "index"). Note that denominator type does NOT affect P-values.

"ses": d for species s is calculated as the standard deviation of RQE values calculated from permuted species weights. This makes the output specificity a standardized effect size (SES). Unfortunately, this makes SES counterintuitively sensitive to occupancy, where species with high occupancy have more extreme SES than rare species, due to their more deterministic sim specificities. Included for comparative purposes, not suggested.

"raw": d is 1 for all species, so output specificity has units of distance, i.e. the raw difference between empirical and simulated RQE. This means that results from different variables are not comparable, since it is not scale-invariant to env or hosts_phylo. It not scale-invariant to the species weights in aunds_mat, either. Not sensitive to number of samples. Not suggested because units are strange, and isn't comparable between variables.

"index": d is the mean of simulated (permuted) RQE values for species that have stronger specificity than expected by chance, resulting in specificity values with range [-1, 0), with 0 as the null hypothesis. In this case, -1 indicates perfect specificity, where a species is associated with zero environmental variability. In the euclidean sense, this could be a species that is always found at the exact same elevation or the exact same pH. For species that have weaker specificity than expected by chance, d is x minus the center (see above) of simulated RQE values, where x is the maximum possible dissimilarity observable given species weights. This d has other useful properties: scale invariance to env/hosts_phylo, insensitivity to the number of samples, insensitivity to occupancy, and strong sensitivity to specificity (default).

`diagnostic`	logical. If true, changes output to include different parts of SES. This includes Pval, SES, raw, denom, emp, and all sim values with column labels as simN where N is the number of sims (default: FALSE)

Value

data.frame where each row is an input species. First column is P-value ($Pval), second column is specificity ($Spec).

Author(s)

John L. Darcy

Examples

```
None yet. Forthcoming examples:
1. calculating regular old elevational specificity the hard way
2. same thing, but using vazquez null model from bipartite package
```

`check_pes_inputs` *check_pes_inputs*

Description

Function used by phy_or_env_spec. checks abunds_mat, env, hosts, and hosts_phylo inputs to phy_or_env_spec to make sure there are no problems. This could include missing species in trees, incompatible dimensions, non-numeric inputs, etc. Returns an input type, which is just a string that can be "mat", "dist", "vec", "phy", or "error".

Usage

```
check_pes_inputs(abunds_mat, env, hosts, hosts_phylo, verbose = TRUE)
```

Arguments

`abunds_mat`	(required, see phy_or_env_spec)
`env`	(required, can be NULL, see phy_or_env_spec)
`hosts`	(required, can be NULL, see phy_or_env_spec)
`hosts_phylo`	(required, can be NULL, see phy_or_env_spec)
`verbose`	logical. Should status messages be displayed? (default: TRUE).

Value

string. either "mat", "dist", "vec", "phy", or "error".

`circularize2dist` *circularize2dist*

Description

Circularizes a vector into a dist object. For example, a vector of days of the year, where the distance between 365 and 2 should be less than the distance between 350 and 365. Another example may be direction, where 0.1 and 2pi radians are close together.

Usage

```
circularize2dist(x, maxx)
```

Arguments

`x`	a numeric vector. All values should be >0.
`maxx`	the maximum theoretical value (also the zero value!) of variable x. In the exampe of months of the year, maxx would be 12, even if you only had data for months 1-8. For degrees, maxx=360. For radians, maxx=2*pi. Must be greater than or equal to values of x.

Value

a vector of differences, ordered identically to a "dist" object.

Author(s)

John L. Darcy

Examples

```
# make some fake data to represent months of the year
months <- c(1, 4, 11)
# run circularize2dist() on the months. Must specify that
# maxx = 12, since december is both 12 and 0 for these data.
circularize2dist(months, 12)
# output is a distance matrix.
# rows and cols of months_circdm are months - it's ordered.
# notice the distance between 11 and 1 is 2, not 10!
```

distcalc *distcalc*

Description

Calculates pairwise geographic distance between locations on earth. Just a convenient wrapper for fields::rdist.earth().

Usage

```
distcalc(lat, lng, sampIDs = NULL)
```

Arguments

`lat`	Numeric vector. Latitudes in decimal degree format.
`lng`	Numeric vector. Longitudes in decimal degree format.
`sampIDs`	Character vector. Sample identifiers. Only required if output dist should have names associated.

Value

matrix containing all pairwise geographic distances in km.

Author(s)

John L. Darcy

Examples

```
data(endophyte)
geo_dists <- distcalc(metadata$Lat, metadata$Lon, metadata$SampleID)
all(rownames(geo_dists) == metadata$SampleID)
```

env_spec_sim *env_spec_sim*

Description

Simulates inputs for phy_or_env_spec, by creating a species distribution over an artificial (or real) environmental variable. That distribution has a mean at the "ideal" environmental value for the simulated species, and the standard deviation of that distribution controls the extent to which the species is specific to the variable. A high SD means weaker specificity, and a low SD means stronger specificity.

Usage

```
env_spec_sim(
  sdev,
  ideal,
  ideal2 = 0,
  ideal3 = 0,
  n_ideal = 1,
  env,
  n_obs,
  up = 0,
  oceanp = 0,
  n_cores = 2,
  seed = 1234567
)
```

Arguments

`sdev`	numeric vector. Standard deviation of the probability distribution P(species), in the same units of env. Low values mean that the species is found acrosss only a narrow range of env, i.e. specificity. High values mean that the species is found across a wide range of env, i.e. cosmopolitanism. Multiple values can be input in order to simulate a range of specificities simultaneously. Can be length 1 or n.
`ideal`	numeric vector. Value of env that is ideal for the simulated species. This is the mode of the probability distribution P(species). Can be length 1 or n.
`ideal2`	numeric vector. Value of env that is the second ideal for the simulated species. Only used if n_ideal >= 2. This is the second mode of the probability distribution P(species). Can be length 1 or n.
`ideal3`	numeric vector. Value of env that is the third ideal for the simulated species. Only used if n_ideal = 3. This is the third mode of the probability distribution P(species). Can be length 1 or n.
`n_ideal`	integer vector. Number of ideal values for the simulated species, i.e. modality of that species' distribution across env; 1 for unimodal, etc. Only can ue values 1, 2, or 3, which correspond to ideal, ideal2, and ideal3. Can be length 1 or n (default: 1).
`env`	numeric vector. Real or fake environmental variable.
`n_obs`	integer vector. Number of positive observations to make, i.e. occupancy of simulated species. Can be length 1 or n (default: 1).
`up`	numeric vector. up=uniform proportion. This is the proportion of the probability distribution P(species) that is composed of a uniform distribution, if desired. If set to a value above zero (and blow 1), P(species) will be a weighted sum of the normal distribution described above, and a uniform distribution. The weight for the uniform distribution will be up, and the weight for the normal distribution will be 1-up (default: 0).
`oceanp`	numeric vector. oceanp=ocean proportion. This is the proportion of samples in env that are "in the ocean", i.e. samples where the species would not expect to

be found even if env is permissive. If aliens were calculating specificity of cows to temperature, they might look in the ocean at sites where the temperature is 17C (great for cows). But cows are not found in the ocean. This proportion is used to randomly select ocean sites within env, and then p(s|env|ocean) = up. Can be length 1 or n (default: 0).

`n_cores` integer. Number of CPU cores for parallel computation (default: 2).

`seed` integer. Seed for randomization. Daughter seeds will be generated for parallel computations, each with the same number of digits as seed (default: 1234567).

Details

Since this process can result in failures (if a species is requested that's highly specific to a region of env that isn't samples), some output species will be failures. default operation is to remove those failures from output matrix and output params data frame, but this can be changed.

Value

List object containing "matrix" and "params" objects:

matrix: matrix where each column is a vector of simulated observation frequencies (counts) corresponding to a value of env; each row represents a simulated species.

params: data.frame of parameters (columns) used to simulate each species (rows).

Author(s)

John L. Darcy

Examples

```
none yet written.
```

`geo_spec_sim` *geo_spec_sim*

Description

Simulates inputs for phy_or_env_spec, by creating a species distribution over artificial (or real) geographic space. That distribution has a bivariate mean at the "ideal" location inspace for the simulated species, and the standard deviation of that (normal) distribution controls the extent to which the species specific to geographic space. A high SD means less specificity, and a low SD means more specificity.

Usage

```
geo_spec_sim(
  sdev,
  n_obs,
  grid,
  ideal_x = 0,
  ideal_y = 0,
  ideal_x2 = 0,
  ideal_y2 = 0,
  ideal_x3 = 0,
  ideal_y3 = 0,
  n_ideal = 1,
  up = 0,
  seed = 123456,
  n_cores = 2
)
```

Arguments

`sdev`	numeric vector. Standard deviation of the probability distribution P(species), in the same units as grid. P(species) is a function of the distance between a sample site and its closest ideal location (specified with ideal_x/2/3 and ideal_y/2/3). Low values mean that the species is found in abundance within only short distances of ideal locations, high values mean the species is found across a wider area. Multiple values can be input in order to simulate a range of specificities simultaneously. Can be length 1 or n.
`n_obs`	integer vector. Number of observations to make, i.e. number of times species is observed. Will be the sum of the species' output column. Can be length 1 or n.
`grid`	data frame with columns x and y, representing cartesian coordinates of sample locations. Can be artificial (generate with randomgrid()) or real.
`ideal_x`	numeric vector. x-coordinate of the ideal spatial location for species (default:0).
`ideal_y`	numeric vector. y-coordinate of the ideal spatial location for species (default:0).
`ideal_x2`	numeric vector. x-coordinate for secondary ideal location. Only used if n_ideal<1 (default:0).
`ideal_y2`	numeric vector. y-coordinate for secondary ideal location. Only used if n_ideal<1 (default:0).
`ideal_x3`	numeric vector. x-coordinate for secondary ideal location. Only used if n_ideal<2 (default:0).
`n_ideal`	integer vector. Number of ideal locations to use. Must be 1, 2, or 3 (default:1).
`up`	numeric vector. up=uniform proportion. This is the proportion of the probability distribution P(species) that is composed of a uniform distribution, if desired. If set to a value above zero (and blow 1), P(species) will be a weighted sum of the normal distribution described above, and a uniform distribution. The weight for the uniform distribution will be up, and the weight for the normal distribution will be 1-up (default: 0).

seed integer. Seed for randomization. Daughter seeds will be generated for parallel computations, each with the same number of digits as seed (default: 1234567).

n_cores integer. Number of CPU cores for parallel computation (default: 2).

Value

List object containing "matrix" and "params" objects:

matrix matrix where each column is a vector of simulated observations for each row in grid; each column of matrix represents a simulated species.

params data.frame of parameters (columns) used to simulate each species (rows).

Author(s)

John L. Darcy

Examples

```
g1 <- randomgrid()
plot(g1)
a1 <- geo_spec_sim(sdev=c(30, 30, 30, 30), n_obs=1000, grid=g1, up=c(0, 0.20, 0.40, 0.60))
par(mfrow=c(2,2))
plot_grid_abunds(g1, a1$matrix[,1])
plot_grid_abunds(g1, a1$matrix[,2])
plot_grid_abunds(g1, a1$matrix[,3])
plot_grid_abunds(g1, a1$matrix[,4])
a2 <- geo_spec_sim(sdev=c(10, 20, 30, 40), n_obs=1000, grid=g1, ideal_x=-50, ideal_x2=50, n_ideal=2)
par(mfrow=c(2,2))
plot_grid_abunds(g1, a2$matrix[,1], main="sd=10")
plot_grid_abunds(g1, a2$matrix[,2], main="sd=20")
plot_grid_abunds(g1, a2$matrix[,3], main="sd=30")
plot_grid_abunds(g1, a2$matrix[,4], main="sd=40")
```

---

make_nested_set *make_nested_set*

Description

Makes a nested set table for a phylo object. Phylo objects made by the ape package store phylogenies as an "adjacency list", which in R is a table within which any given edge is represented by the two node numbers it connects. With this data structure, it is very computationally expensive to figure out which tips are the descendents of a given node. Instead, using a "nested set" data structure, this operation is trivial. A nested set stores the minimum and maximum tip index for each node, such that the descendents of that node are given by the inclusive range between those values.

Usage

```
make_nested_set(phy, n_cores = 2)
```

Arguments

phy	phylo object. Must be rooted, and sorted such that tip indices are ordered. This is the default for rooted trees read in using ape's read.tree function.
n_cores	integer. Number of CPU cores to use (DEFAULT: 2). lapply will be used instead of mclapply if ncores is 1.

Value

Matrix object representing a nested set of nodes. Each row matches rows of the "edges" object within phy. Object has the following columns:

1 (node) Node value in the original phylo object.

2 (min) minimum tip index subtended by node.

3 (max) maximum tip index subtended by node.

4 (contig) Is min:max congiguous? 1 (true) or 0 (false).

Author(s)

John L. Darcy

References

https://en.wikipedia.org/wiki/Nested_set_model https://en.wikipedia.org/wiki/Adjacency_list

See Also

ape::phylo

Examples

```
library(specificity)
phy <- get(data(endophyte))$supertree
# check if tree is rooted:
is.rooted(phy)
# make nested set table:
phy_ns <- make_nested_set(phy)
# show that nested set table matches up with edges table in phy:
all(phy$edge[,2] == phy_ns[,1])
```

occ_threshold *occ_threshold*

Description

removes species (columns) from a matrix that don't meet a minimum occupancy, defined as the number of samples in which that species was observed.

Usage

```
occ_threshold(m, threshold, max_absent = 0)
```

Arguments

m	matrix or data frame of numeric values. Columns represent species, rows are samples.
threshold	integer. Minimum number of samples a species can occupy without being removed.
max_absent	float. Maximum abundance value at which a species will be considered absent (default: 0).

Value

matrix with low-occupancy species removed.

Author(s)

John L. Darcy

Examples

```
attach(endophyte)
dim(zotutable)
zotutable_over25 <- occ_threshold(zotutable, 25)
dim(zotutable_over25)
```

`onto2nwk` *onto2nwk*

Description

Converts an ontology (higherarchical categories) into a nwk phylogeny.

Usage

```
onto2nwk(df)
```

Arguments

`df`	a data.frame object where columns represent ontology levels, which are assumed to be nested hierarchically. this function does not check for proper hierarchical nestedness - it is the user's job to check that each node and tip name is monophyletic. Lower levels (e.g. tips) should be the rightmost column of df, and higher levels (e.g. roots) should be leftmost column, with intermediate columns ordered between.

Value

A newick (nwk) format string.

Author(s)

John L. Darcy

Examples

```
df <- data.frame(
l1 = c( "a", "a", "a", "a", "a", "a", "a", "b", "b", "b", "b", "b", "b", "c", "d"),
l2 = c( "e", "e", "e", "e", "f", "f", "g", "h", "h", "h", "i", "j", "j", "k", "l"),
l3 = c( "m", "n", "o", "o", "p", "p", "q", "r", "r", "s", "t", "u", "v", "w", "x")
)
nwk_str <- onto2nwk(df)
a <- ape::read.tree(text=nwk_str)
plot(a, show.node.label=TRUE)
```

`pairwise_product` *pairwise_product*

Description

Calculates pairwise_products from unique 2-element combinations of vector x. The output vector is the same length and same order as a lower triangle of matrix with rows and columns x.

Usage

```
pairwise_product(x)
```

Arguments

x	numeric vector.

Value

vector of pairwise_products, of length (l^2-l)/2, where l=length(x).

Author(s)

John L. Darcy

Examples

```
x <- 1:6
y_cpp <- pairwise_geo_mean(x)
y_r <- as.dist(outer(x, x, function(x,y){x*y}))
print("Calculated with R's outer() function:")
y_r
print("As a vector:")
as.vector(y_r)
print("Calculated with pairwise_product (C++):")
y_cpp
```

`phy_or_env_spec` *phy_or_env_spec*

Description

Calculates species' specificities to either a 1-dimensional variable (vector), 2-dimensional variable (matrix), or to a phylogeny. Transforms all variable input types into a matrix D, and calculates specificity by comparing empirical Rao's Quadratic Entropy to simulated RQE (same but with permuted abundances). By default (denom_type = "index"), an index is calculated from emp and sim values such that Spec=0 indicates random assortment (null hypothesis), and more negative values indicate stronger specificity.

Usage

```
phy_or_env_spec(
  abunds_mat,
  env = NULL,
  hosts = NULL,
  hosts_phylo = NULL,
  n_sim = 1000,
  p_adj = "fdr",
  seed = 1234567,
  tails = 1,
  n_cores = 2,
  verbose = TRUE,
  p_method = "raw",
  center = "mean",
  denom_type = "index",
  diagnostic = F
)
```

Arguments

abunds_mat	matrix or data frame of numeric values. Columns represent species, rows are samples. For columns where the value is nonzero for two or fewer data points, specificity cannot be calculated, and NAs will be returned. Negative values in abunds_mat are not allowed (REQUIRED).
env	numeric vector, dist, or square matrix. Environmental variable corresponding to abunds. For example, temperature, or geographic distance. Not required for computing phylogenetic specificity (default: NULL).
hosts	character vector. Host identities corresponding to abunds. Only required if calculating phylogenetic specificity (default: NULL).
hosts_phylo	phylo object. Tree containing all unique hosts as tips. Only required if calculating phylogenetic specificity (default: NULL).
n_sim	integer. Number of simulations of abunds_mat to do under the null hypothesis that host or environmental association is random. P-values will not be calculated if n_sim < 100 (default: 500).
p_adj	string. Type of multiple hypothesis testing correction performed on P-values. Can take any valid method argument to p.adjust, including "none", "bonferroni", "holm", "fdr", and others (default: "fdr").
seed	integer. Seed to use so that this is repeatable (default: 1234557).
tails	integer. 1 = 1-tailed, test for specificity only. 2 = 2-tailed. 3 = 1-tailed, test for cosmopolitanism only. 0 = no test, P=1.0 (default: 1).
n_cores	integer. Number of CPU cores to use for parallel operations. If set to 1, lapply will be used instead of mclapply (default: 2).
verbose	logical. Should status messages be displayed? (default: TRUE).
p_method	string. "raw" for quantile method, or "gamma_fit" for calculating P by fitting a gamma distribution (default: "raw").

`center` string. Type of central tendency to use for simulated RQE values. Options are "mean", "median", and "mode". If mode is chosen, a reversible gamma distribution is fit and mode is calculated using that distribution (default: mean).

`denom_type` string. Type of denominator (d) to use (default: "index"). Note that denominator type does NOT affect P-values.

"ses": d for species s is calculated as the standard deviation of RQE values calculated from permuted species weights. This makes the output specificity a standardized effect size (SES). Unfortunately, this makes SES counterintuitively sensitive to occupancy, where species with high occupancy have more extreme SES than rare species, due to their more deterministic sim specificities. Included for comparative purposes, not suggested.

"raw": d is 1 for all species, so output specificity has units of distance, i.e. the raw difference between empirical and simulated RQE. This means that results from different variables are not comparable, since it is not scale-invariant to env or hosts_phylo. It not scale-invariant to the species weights in aunds_mat, either. Not sensitive to number of samples. Not suggested because units are strange, and isn't comparable between variables.

"index": d is the mean of simulated (permuted) RQE values for species that have stronger specificity than expected by chance, resulting in specificity values with range [-1, 0), with 0 as the null hypothesis. In this case, -1 indicates perfect specificity, where a species is associated with zero environmental variability. In the euclidean sense, this could be a species that is always found at the exact same elevation or the exact same pH. For species that have weaker specificity than expected by chance, d is x minus the center (see above) of simulated RQE values, where x is the maximum possible dissimilarity observable given species weights. This d has other useful properties: scale invariance to env/hosts_phylo, insensitivity to the number of samples, insensitivity to occupancy, and strong sensitivity to specificity (default).

`diagnostic` logical. If true, changes output to include different parts of Spec. This includes Pval, Spec, raw, denom, emp, and all sim values with column labels as simN where N is the number of sims (default: FALSE)

Value

data.frame where each row is an input species. First column is P-value ($Pval), second column is specificity ($Spec).

Author(s)

John L. Darcy

References

- Poulin et al. (2011) Host specificity in phylogenetic and geographic space. Trends Parasitol 8:355-361. doi: 10.1016/j.pt.2011.05.003
- Rao CR (2010) Quadratic entropy and analysis of diversity. Sankhya 72:70-80. doi: 10.1007/s13171-010-0016-3

- Rao CR (1982) Diversity and dissimilarity measurements: A unified approach. Theor Popul Biol 21:24-43.

Examples

```
# phylogenetic specificity using endophyte data set
attach(endophyte)
# only analyze species with occupancy >= 20
specs_list <- list()
m <- occ_threshold(prop_abund(zotutable), 20)
specs_list$host <- phy_or_env_spec(
    abunds_mat=m,
    hosts=metadata$PlantGenus,
    hosts_phylo=supertree,
    n_cores=12
)

# environmental specificity using elevation from endophyte data set:
specs_list$elev <- phy_or_env_spec(
    abunds_mat=m,
    env=metadata$Elevation,
    n_cores=12
)

# geographic specificity using spatial data from endophyte data set:
specs_list$geo <- phy_or_env_spec(
    abunds_mat=m,
    env=distcalc(metadata$Lat, metadata$Lon),
    n_cores=12
)

plot_specificities(specs_list)
```

---

phy_spec_sim *phy_spec_sim*

Description

Simulates inputs for phy_or_env_spec, by creating a species distribution over an artificial (or real) host phylogenetic tree. For a phylogeny, the species probability distribution P(s) is based on patristic distances within the tree, such that P(s) is maximized at zero patristic distance between a tip in the tree and the ideal host species for s. This distribution is given by a a truncated normal distribution centered on zero, using only positive values. A uniform proportion (up) to that distribution may be added as well, to add a baseline probability to P(s). The standard deviation of P(s) can be raised or lowered to simulate cosmopolitanism or specificity.

Usage

```
phy_spec_sim(
  sdev,
  ideal,
  ideal2 = "",
  ideal3 = "",
  n_ideal = 1,
  hosts,
  hosts_phylo,
  n_obs,
  up = 0,
  oceanp = 0,
  n_cores = 2,
  seed = 1234567
)
```

Arguments

`sdev`	numeric vector. Standard deviation of the probability distribution P(s), in units of patristic distance in hosts_phylo. Low values mean that species s is found with a narrow grouping of hosts, i.e. specificity. High values mean that s is found across a wider group of hosts, i.e. cosmopolitanism. Multiple values can be input in order to simulate a range of specificities, simultaneously. To get a handle on this somewhat opaque variable, consider plotting a histogram of patristic distances within hosts_phylo (see: ape::cophenetic.phylo). Can be length 1 or n.
`ideal`	character vector. Tip label of hosts_phylo that is ideal (or closest to ideal) for the simulated species. Does not have to be in hosts, but MUST be in hosts_phylo. Can be length 1 or n.
`ideal2`	character vector. Tip label of hosts_phylo that is secondary ideal host for the simulated species. Does not have to be in hosts, but MUST be in hosts_phylo. Can be blank ("") if corresponding n_ideal < 2. Can be length 1 or n (default: "").
`ideal3`	character vector. Tip label of hosts_phylo that is tertiary ideal host for the simulated species. Does not have to be in hosts, but MUST be in hosts_phylo. Can be blank ("") if corresponding n_ideal < 3. Can be length 1 or n (default: "").
`n_ideal`	integer vector. number of ideal hosts to use. Must be 1, 2, or 3 (default: 1).
`hosts`	character vector. Real of fake host identities. All must be tips within hosts_phylo. Analogous to env argument to env_spec_sim.
`hosts_phylo`	phylo object. Tree containing all unique hosts as tips.
`n_obs`	integer vector. Number of positive observations to make, i.e. occupancy of simulated species. Can be length 1 or n.
`up`	numeric vector. up=uniform proportion. This is the proportion of the probability distribution P(species) that is composed of a uniform distribution, if desired. If set to a value above zero (and blow 1), P(species) will be a weighted sum of the normal distribution described above, and a uniform distribution. The weight for

	the uniform distribution will be up, and the weight for the normal distribution will be 1-up (default: 0).
`oceanp`	numeric vector. See ?env_spec_sim for help.
`n_cores`	integer. Number of CPU cores for parallel computation (default: 2).
`seed`	integer. Seed for randomization. Daughter seeds will be generated for parallel computations, each with the same number of digits as seed (default: 1234567).

Value

List object containing "matrix" and "params" objects:

matrix: matrix where each column is a vector of simulated observations corresponding to a value of hosts; each row represents a simulated species.

params: data.frame of parameters (columns) used to simulate each species (rows). A column called "index" is included so that simulated species can be mapped back onto original data structures when some species are ommitted due to simulation failure (see fail_rm).

Author(s)

John L. Darcy

Examples

```
none yet written.
```

`plot_grid_abunds` *plot_grid_abunds*

Description

plots species abundances across spatial sampling locations

Usage

```
plot_grid_abunds(grid, abunds, pch = "", ...)
```

Arguments

`grid`	data frame with columns x and y, representing cartesian coordinates of sample locations. Can be artificial (generate with randomgrid()) or real.
`abunds`	abundances of a species, corresponding to rows in grid.
`pch`	pch character code to use for bottom of each abundance line (default: "")
`...`	arguments to be passed to plot.

Value

returns nothing, just makes a plot.

Author(s)

John L. Darcy

Examples

```
g1 <- randomgrid()
plot(g1)
a1 <- geo_spec_sim(sdev=c(30, 30, 30, 30), n_obs=1000, grid=g1, up=c(0, 0.20, 0.40, 0.60))
par(mfrow=c(2,2))
plot_grid_abunds(g1, a1$matrix[,1])
plot_grid_abunds(g1, a1$matrix[,2])
plot_grid_abunds(g1, a1$matrix[,3])
plot_grid_abunds(g1, a1$matrix[,4])
```

---

plot_pairwise_spec *plot_pairwise_spec*

Description

Plots pairwise correlations between specificity to multiple variables. Specificity results are supplied to this function as a list of specificity tables, i.e. a list where each object within the list is an output of phy_or_env_spec, and all were created using the same abunds_mat object (see: ?phy_or_env_spec).

Usage

```
plot_pairwise_spec(
  sl,
  label_cex = 1,
  point_cex = 1,
  cor_cex = 2,
  cor_red_lim = 0.7,
  method = "pearson"
)
```

Arguments

sl	"specs list" list of outputs from phy_or_env_spec as described above.
label_cex	float. Size of variable labels, which will be displayed along the plot's diagonal. Use cex units; see ?par (default: 1).
point_cex	float. Size of points in the plot's lower triangle. Useful to reduce this if you are plotting lots of species. Use cex units; see ?par (default: 1).

cor_cex	float. Size of text for correlations displayed in plot's upper triangle. Use cex units; see ?par (default: 1).
cor_red_lim	float. Correlation coefficients will be shown in red if they are equal to or more extreme than this value (default: 0.70).
method	string. Preferred correlation method. see ?cor for options (default: "pearson").

Value

Returns nothing. Plots correlations in a square matrix of subplots, where variable names are shown in the diagonal, pairwise specificities are plotted in the lower triangle, and correlation coefficients are displayed in the upper triangle. For plots in the lower triangle, each point represents a species.

Author(s)

John L. Darcy

Examples

```
library(specificity)
attach(endophyte)
otutable_over10 <- occ_threshold(otutable, threshold = 10)
specs_list <- list()
specs_list$NDVI <- phy_or_env_spec(otutable_over10, env=metadata$NDVI,
  n_cores=10, n_sim=100, p_method="gamma_fit")
specs_list$Evapotranspiration <- phy_or_env_spec(otutable_over10,
  env=metadata$Evapotranspiration, n_cores=10, n_sim=100, p_method="gamma_fit")
specs_list$Rainfall <- phy_or_env_spec(otutable_over10, env=metadata$Rainfall,
  n_cores=10, n_sim=100, p_method="gamma_fit")
plot_pairwise_spec(specs_list)
```

---

plot_specs_stacks *plot_specs_stacks*

Description

Visualizes results from phy_or_env_spec as stacked histograms. Aliased to plot_specificities() for backward compatibility.

Usage

```
plot_specs_stacks(
  specs_list,
  n_bins = 20,
  col_sig = "black",
  col_nsig = "gray",
  col_bord = NA,
```

```
  alpha = 0.05,
  label_cex = 1
)
```

Arguments

`specs_list`	list of data.frames. Each data.frame must be an output from phy_or_env_spec; must have columns "Spec" and "Pval".
`n_bins`	integer. Number of bins for stacked violins (default: 20).
`col_sig`	string. Color name or hex code for species where Pval <= alpha (default: "black").
`col_nsig`	string. Color name or hex code for species where Pval > alpha (default: "gray").
`col_bord`	string. Color name or hex code for border color. Use NA for no border (default: NA).
`alpha`	float. alpha value for determining statistical significance; see col_sig and col_nsig above (default: 0.05).
`label_cex`	float. Used to change size of x-axis labels (default: 1).

Value

returns nothing (a plot is made).

Author(s)

John L. Darcy

Examples

```
data(endophyte); attach(endophyte)
otutable <- occ_threshold(prop_abund(otutable), 10)
elev_spec <- phy_or_env_spec(otutable, env=metadata$Elevation, n_sim=100, p_method="gamma_fit")
ndvi_spec <- phy_or_env_spec(otutable, env=metadata$NDVI, n_sim=100, p_method="gamma_fit")
plot_specs_stacks(list(Elevation=elev_spec, NDVI=ndvi_spec))
```

`plot_specs_violin` *plot_specs_violin*

Description

Visualizes results from phy_or_env_spec as violins. Violin area is proportionally divided such that lighter colors represent density of non-significant features, and darker colors represent statistically significant features.

Usage

```
plot_specs_violin(
  specs_list,
  cols = "black",
  cols_bord = "white",
  alpha = 0.05,
  label_cex = 1,
  nsig_trans = 0.3,
  minval = -1,
  maxval = 1,
  ylab = "Spec",
  ...
)
```

Arguments

`specs_list`	list of data.frames. Each data.frame must be an output from phy_or_env_spec; must have columns "Spec" and "Pval".
`cols`	character vector of color names or hex codes. If only one value is given, all violins will be that color. Otherwise, one value may be given per item in specs_list, corresponding to its order (default: "black").
`cols_bord`	character vector of color names or hex codes. Color name or hex code for borders drawn around and within violins. Length 1 or length n, just like cols. For no borders, use cols_bord=NA (default: "white").
`alpha`	float. alpha value for determining statistical significance (default: 0.05).
`label_cex`	float. Used to change size of x-axis labels (default: 1).
`nsig_trans`	float between 0 and 1 (inclusive). Determines how transparent violin area will be for nonsignificant features, with 0 meaning totally transparent and 1 meaning totally opaque (default: 0.4).
`minval`	minimum possible value for specificity statistic (default: -1).
`maxval`	maximum possible value for specificity statistic (default: 1).
`ylab`	y-axis label for plot (default:"Spec").
`...`	additional arguments to be passed to polygon().

Value

returns nothing (a plot is made).

Author(s)

John L. Darcy

Examples

```
data(endophyte); attach(endophyte)
otutable <- occ_threshold(prop_abund(otutable), 10)
elev_spec <- phy_or_env_spec(otutable, env=metadata$Elevation,
  n_sim=100, p_method="gamma_fit")
ndvi_spec <- phy_or_env_spec(otutable, env=metadata$NDVI,
  n_sim=100, p_method="gamma_fit")
# default black
plot_specs_violin(list(Elevation=elev_spec, NDVI=ndvi_spec))
# with colors
plot_specs_violin(list(Elevation=elev_spec, NDVI=ndvi_spec),
  cols=c("orange", "forestgreen"))
# with border colors
plot_specs_violin(list(Elevation=elev_spec, NDVI=ndvi_spec),
  cols=c("orange", "forestgreen"),
  cols_bord=c("blue", "red"))
# with thicker borders (arg "lwd" is passed to polygon())
plot_specs_violin(list(Elevation=elev_spec, NDVI=ndvi_spec),
  cols=c("orange", "forestgreen"), cols_bord="black", lwd=3)
```

---

prop_abund *prop_abund*

Description

Calculates proportional abundance of each species (columns) across samples (rows) in community data matrix m. Row sums of output matrix will all be 1.

Usage

```
prop_abund(
  m,
  to_int = FALSE,
  max_int = floor(sqrt(.Machine$integer.max)),
  speciesRows = FALSE
)
```

Arguments

m	matrix or data frame of numeric values. Columns represent species, rows are samples.
to_int	logical. Should output matrix be transformed into integers from 0 to max_int? Integers take up half as much space as doubles, and as weights are equivalent for calculating specificity. The tradeoff is a little bit of precision (default: FALSE).
max_int	integer. Maximum integer value used for to_int. If pairwise geometric means will be calculated with these data, it is nice to keep this value as the square root of the maximum integer size, which is the default.
speciesRows	logical. Do rows represent species (instead of samples)? (default:FALSE)

Value

matrix of proportional abundances.

Author(s)

John L. Darcy

Examples

```
attach(endophyte)
m_dbl <- prop_abund(zotutable)
m_int <- prop_abund(zotutable, to_int=TRUE)
head(rowSums(m_dbl))
head(rowSums(m_int))
# note that they are off by a little bit. This small loss in precision is OK.
object.size(m_dbl)
object.size(m_int)
random_positions <- random_rep_positions(m_dbl, 100)
plot(m_int[random_positions] ~ m_dbl[random_positions])
```

---

randomgrid *randomgrid*

Description

Generates a random spatial sampling using a bivariate random uniform distribution.

Usage

```
randomgrid(
  n_samp = 1000,
  xmin = -100,
  xmax = 100,
  ymin = -100,
  ymax = 100,
  seed = 123456
)
```

Arguments

n_samp	number of sampling locations to output (default: 1000).
xmin	minimum x-axis coordinate (default: -100).
xmax	maximum x-axis coordinate (default: 100).
ymin	minimum y-axis coordinate (default: -100).
ymax	maximum y-axis coordinate (default: 100).
seed	integer, seed for randomization.

Value

data.frame object with x and y columns, with n_samp rows.

Author(s)

John L. Darcy

Examples

```
g <- randomgrid()
plot(g)
g2 <- randomgrid(nsamp=50, xmin=0, ymin=0)
plot(g2)
```

`random_rep_positions` *random_rep_positions*

Description

Finds positions in a vector (or matrix) that are randomly located within n_bins evenly sized bins. This is useful for 1:1 comparisons of large vectors where plotting or comparing all points is prohibitive. Only used in an example for the prop_abund() function.

Usage

```
random_rep_positions(x, nbins = 50)
```

Arguments

`x`	vector
`nbins`	number of bins to use

Value

integer vector of positions that were selected

Author(s)

John L. Darcy

`rao1sp` *rao1sp*

Description

Calculate's Rao's quadratic entropy for one species (rao1sp = "rao one species").

Usage

```
rao1sp(p, D, perm)
```

Arguments

`p`	numeric vector of length n - a species weights vector.
`D`	numeric vector of length n(n-1)/2 - i.e. a dist object whose full matrix is nxn.
`perm`	bool. Whether or not the permute order of p before calculating Rao.

Value

A single Rao value.

Author(s)

John L. Darcy

Examples

```
# p <- 1:5/sum(1:5)
# D <- dist(6:10)
# rao1sp(p, D)
```

`raoperms` *raoperms*

Description

C++ function used by phy_or_env_spec(). Not meant for use otherwise.

Usage

```
raoperms(p,D,nsim,seed)
```

Arguments

p	numeric vector of species weights.
D	numeric vector (dist) of distances corresponding to a lower triangle of a matrix whose rows and cols correspond to p; i.e. an lxl matrix where l is length(p). R's dist() function does this for you!
nsim	integer. number of sims to do
seed	integer. seed for randomization

Value

Vector of Rao values, length = nsim.

Author(s)

John L. Darcy

`rao_genetic_max` *rao_genetic_max*

Description

Uses a genetic algorithm to find the optimum permutation of p to maximize Rao(p,D).

Usage

```
rao_genetic_max(p, D)
```

Arguments

p	numeric vector of length n - a species weights vector.
D	numeric vector of length n(n-1)/2 - i.e. a dist object whose full matrix is nxn.
term_cycles	integer, number of cycles with no improvement to trigger termination (DEFAULT=10).
maxiters	integer, maximum number of iterations to run algorithm (DEFAULT=400).
popsize	integer, population size for genetic algorithm (DEFAULT=300).
keep	integer, number of individuals to keep during each iteration (DEFAULT=5).
prc	double, precision for calculating termination with term_cycles (DEFAULT=0.001).

Value

List object containing results of genetic algorithm:

best_rao: Maximum Rao value found.

iter_raos: Max Rao value for each iteration. If termination condition was met, rest of values after final iteration are NA.

iterations: Iteration numbers, corresponding to iter_raos.

best_p: The best permutation of p found (corresponds to best_rao).

Author(s)

John L. Darcy

Examples

```
set.seed(12345)
p <- runif(100) * 20
D <- dist(sample(p))
a <- rao_genetic_max(p,D)
plot(a$iter_raos ~ a$iterations)
```

---

rao_sort_max *rao_sort_max*

Description

Sorts pairwise_product(p) and D to approximate the maximum of Rao(p,D) under permutations of p.

Usage

```
rao_sort_max(p, D)
```

Arguments

p	numeric vector of length n - a species weights vector.
D	numeric vector of length n(n-1)/2 - i.e. a dist object whose full matrix is nxn.

Value

A single value, approximating maximum rao under permutations of p.

Author(s)

John L. Darcy

Examples

```
# data(endophyte)
# p <- prop_abund(endophyte$otutable)[,1]
# D <- dist(endophyte$metadata$Elevation)
# rsm <- rao_sort_max(p,D)
```

`tips_from_node` *tips_from_node*

Description

Determines which tip indices in a phylogeny descend from a given node. Called by make_nested_set(), not intended for use otherwise, but some may find it handy. Data should come from a rooted phylogeny, but this function doesn't check that so be careful.

Usage

```
tips_from_node(nodes, anc, des)
```

Arguments

`nodes`	integer vector or scalar. The node index or indices for which tip indices are desired.
`anc`	integer vector. "ancestor" column vector from an adjacency matrix. For an ape::phylo object phy, anc=phy$edge[,1].
`des`	integer vector. "descendant" column vector from an adjacency matrix. For an ape::phylo object phy, des=phy$edge[,2].

Value

integer vector of tip indices, in no particular order.

Author(s)

John L. Darcy

See Also

ape::phylo

Examples

```
library(specificity)
phy <- get(data(endophyte))$supertree
# check if tree is rooted:
is.rooted(phy)
# which tips are in the Cucurbitales?
plot(phy) # need to stretch out the plot to see...
nodelabels(adj=c(0,-1), bg="yellow") # node numbers
nodelabels(phy$node.label, adj=c(0,1), bg="lightblue") # node names
# we can see that Cucurbitales is node 107
cuc_tips <- tips_from_node( nodes=107, anc=phy$edge[,1], des=phy$edge[,2] )
cuc_tips
phy$tip.label[cuc_tips]
```

tree2mat *tree2mat*

Description

Transforms a phylogenetic tree into a dist object containing patristic distances between tips. Dists are just lower triangles of matrices, and the rows and columns of that matrix are defined by a user-supplied vector of tip labels, which can include duplicate values. Contrast with ape::cophenetic.phylo, which produces a distance matrix containing only unique pairwise patristic distances within the phylogeny.

Usage

```
tree2mat(tree, x, n_cores = 1, delim = ";")
```

Arguments

`tree`	phylo object. Tree containing all unique species in x as tips. May contain tips that are not in x.
`x`	character vector. Vector of species identities, each of which must be in tree as a tip label. May contain any given species identity more than once.
`n_cores`	integer. Number of cores to use for parallel computation. No parallelization will be done if n_cores = 1. Multithreading should only be used for large trees where x has low redundancy (default: 1).
`delim`	string. Delimiter character or string for internal use. Must not be present in tree$tip.label. This is checked by the function and will return an error otherwise (default: ";").

Value

dist object, of vector length equal to (l^2-l)/2 where l is length(x); i.e. values are the lower triangle of a patristic distance matrix with rows=x and cols=x.

Author(s)

John L. Darcy

Examples

```
example_tree <- ape::read.tree(text=" ((((a:1,b:1):1,c:2):1,d:3):1,(e:1,f:1):3);")
example_x <- c("a", "a", "a", "b", "c", "d", "c", "a", "f")
# unique patristic distance matrix:
ape::cophenetic.phylo(example_tree)
# dist object for example_x:
tree2mat(tree=example_tree, x=example_x)

# examples with other delimiters
```

```
tree2mat(tree=example_tree, x=example_x, delim="@")
tree2mat(tree=example_tree, x=example_x, delim="i love cats")
# should fail since "a" is in a tip name:
tree2mat(tree=example_tree, x=example_x, delim="a")
```

---

wpd *wpd*

Description

Calculates weighted Phylogenetic Diversity for a vector s of species observations, weighted by the frequency of each species within s. For example, if s=a, a, b, a, b, c, a, then species a will have weight 4, species b will have weight 2, and species c will have weight 1. Unobserved species have weight zero. However, one may wish to exclude observations that do not meet some criterion, such as co-observation of a symbiote or parasite. For this reason, a second set of weights w can be provided as a vector of numeric values that are paired with s. These weights are then implicitely combined with the weights discussed above depending on which weighted metric is chosen. In the case of Phylogenetic Entropy (Hw), per-tip weights are calculated as the sums of w. In the case of Weighted Faith (WF), per-tip weights are averages of w.

Usage

```
wpd(s, s_phylo, w = NULL, nested_set = NULL, metric = "Hp")
```

Arguments

`s`	character vector. One species name per observation. If no species was observed for a given datum, use NA. s can also be provided as a vector of unique species identities, in which case counts of those species can be given as w.
`s_phylo`	phylo object. Tree containing all unique names in s as tips. Must not contain duplicate tip labels.
`w`	numeric vector. Optional weights for s, e.g. number of parasites observed in each sample, or boolean weights corresponding to presence or absence of parasite species, or confidence species was observed, etc. If w is not provided but a weighted metric is specified, w will be set to 1 for each value of s. Thus, weights for each unique species in s would be equal to the number of times that species appears in s. w is not used for unweighted metrics (PD). Any NA values in w will be pairwise removed from w and s (default: NULL).
`nested_set`	matrix. The output of make_nested_set(s_phylo). If not provided, will be calculated on the fly. Precalculation only provides speedup with very large trees (default: NULL).
`metric`	character. Abbreviated name of desired tree-based phylogenetic diversity metric. Available metrics are: **Hp:** Phylogenetic Entropy. Insensitive to 0 weights, cannot increase with removal of taxa. Allen et al. 2009.

WF: Weighted Faith's PD. Sensitive to 0 weights, i.e. a clade that was heavily sampled but has lots of zeroes will cause its sister clades to be underrepresented. Swenson 2014.

PD: Original Faith's Phylogenetic Diversity. Unweighted. Simply a sum of branch- lengths in your tree (but only for taxa in s). Faith 1992.

Value

Single WPD or PD value.

Author(s)

John L. Darcy

References

- Allen B, Kon M, Bar-Yam Y (2009) A new phylogenetic diversity measure generalizing the Shannon index and its application to Phyllostomid bats. American Naturalist 174(2).
- Swenson NG (2014) Functional and Phylogenetic Ecology in R. Springer UseR! Series, Springer, New York, New York, U.S.A.
- Faith DP (1992) Conservation evaluation and phylogenetic diversity. Biological Conservation 61.

Examples

```
library(specificity)
set.seed(12345)
s_phylo <- get(data(endophyte))$supertree
w <- sample(c(0, 1), replace=T, size=10)
s <- sample(s_phylo$tip.label, replace=T, size=10)
wpd(s, s_phylo, w, metric="Hp")
```

`wpd_table` *wpd_table*

Description

Calculates phylogenetic entropy (Hp) for each column vector s of species observations within matrix m, weighted by the frequency of each species within s. Can also calculate Faith's PD.

Usage

```
wpd_table(m, s_phylo, nested_set, metric = "Hp", ncores = 4)
```

Arguments

`m`	matrix of species observation vectors (s). See s argument of wpd().
`s_phylo`	phylo object. Tree containing all unique names in s as tips. Must not contain duplicate tip labels.
`nested_set`	matrix. The output of make_nested_set(s_phylo). If not provided, will be calculated on the fly. Precalculation only provides speedup with very large trees (default: NULL).
`metric`	character. Abbreviated name of desired tree-based phylogenetic diversity metric. Available metrics are: **Hp:** Phylogenetic Entropy. Insensitive to 0 weights, cannot increase with removal of taxa. Allen et al. 2009. **WF:** Weighted Faith's PD. Sensitive to 0 weights, i.e. a clade that was heavily sampled but has lots of zeroes will cause its sister clades to be underrepresented. Swenson 2014. **PD:** Original Faith's Phylogenetic Diversity. Unweighted. Simply a sum of branch- lengths in your tree (but only for taxa in s). Faith 1992.

Value

multiple WPD or PD values, one for each column of m.

Author(s)

John L. Darcy

References

- Allen B, Kon M, Bar-Yam Y (2009) A new phylogenetic diversity measure generalizing the Shannon index and its application to Phyllostomid bats. American Naturalist 174(2).
- Swenson NG (2014) Functional and Phylogenetic Ecology in R. Springer UseR! Series, Springer, New York, New York, U.S.A.
- Faith DP (1992) Conservation evaluation and phylogenetic diversity. Biological Conservation 61.

Examples

```
none yet written.
```

Index