# Package 'specificity'

August 30, 2019

**Title** Calculate Environmental or Host Phylogenetic Specificity

**Version** 0.0.0.9000

**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. The mathematics for the standardized effect size of phylogenetic specificity were adapted from Poulin et al. (2011). See: Poulin R, Krasnov BR, Mouillot D (2011) Host specificity in phylogenetic and geographic space. Trends in Parasitology 27(8): 355-361.

**License** GPL

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Depends** ape, geiger, parallel

**Imports** Rcpp, fields, testthat

**LinkingTo** Rcpp

**NeedsCompilation** yes

**Author** John L Darcy [aut, cre]

**Maintainer** John L Darcy <darcyj@colorado.edu>

## 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
```

---

# daughter_seeds *daughter_seeds*

---

## Description

Makes n daughter seeds from seed s. This is useful for processes one wishes to be deterministic, but may not be executed in the same order every time.

## Usage

```
daughter_seeds(n, s = 12345)
```

## Arguments

| | |
|---|---|
| n | integer. Number of daughter seeds to make. |
| s | integer. A seed (DEFAULT: 12345). |

## Value

vector of length n containing integer seeds.

## Author(s)

John L. Darcy

## distcalc *geo_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 less specificity, and a low SD means more 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. |
| `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).

`ideal_x3` numeric vector. x-coordinate for secondary ideal location. Only used if n_ideal<2 (DEFAULT=0).

## 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 geiger::tips

## Examples

```
library(geiger)
library(ape)
library(parallel)
phy <- get(data(geospiza))$phy
# check if tree is rooted:
is.rooted(phy)
# make nested set table:
phy_ns <- make_nested_set(phy)
# show that nested set table matches up with edges table in phy:
all(phy$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)
```

---

## pairwise_geo_mean *pairwise_geo_mean*

---

### Description

Calculates pairwise geometric means from unique 2-element combinations of vector x. Written in C++ because R slow. The output vector is the same length and same order as a lower triangle of matrix with rows and columns x.

### Usage

```
pairwise_geo_mean(x)
```

### Arguments

| | |
|---|---|
| x | numeric vector. |

### Value

vector of pairwise geometric means, of length (l*l-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){sqrt(x*y)}))
print("Calculated with R's outer() function:")
y_r
print("As a vector:")
as.vector(y_r)
print("Calculated with pairwise_geo_mean (C++):")
y_cpp
```

## `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 standardized effect size (SES) of species' specificity to either a 1-dimensional variable (vector), 2-dimensional variable (matrix), or to a phylogeny. Default operation is to transform all variable input types into a matrix M, and calculate specificity as SES of the empirical sum of M weighted by weights matrix W. W is a matrix of pair-wise products of species species abundances (Rao's quadratic entropy). Optionally, host phylogenetic specificity can be calculated per Poulin et al. (2011) using weighted phylogenetic entropy, and specificity to 1-dimensional data can be calculated using a weighted variance approach. These approaches are included mainly for comparative methodological purposes.

## Usage

```
phy_or_env_spec(abunds_mat, env = NULL, hosts = NULL,
  hosts_phylo = NULL, n_sim = 1000, sim_fun = function(m) {
  m[sample(1:nrow(m)), ] }, p_adj = "fdr", seed = 1234567, tails = 1,
  n_cores = 2, verbose = TRUE, lowmem = FALSE, vec_wvar = FALSE,
  phy_ent = FALSE, denom_type = "global_unif")
```

## 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, environmental SES 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 SES for phylogenetic specificity (DEFAULT: NULL). |
| `hosts_phylo` | phylo object. Tree containing all unique hosts as tips. Only required if calculating SES for 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). |
| `sim_fun` | function. A function f where f(abunds_mat) returns a matrix object with the same number of rows and columns as abunds_mat. Default is f=function(m) m[sample(1:nrow(m)),], which just permutes the order of rows in abunds_mat. Users may wish to use a null model that is able to preserve row and column totals such as the function permatswap() from the vegan package or the function vaznull() from the bipartite package. Either of these can be easily adapted to return only a single matrix (see examples). However, neither can accomodate non-integer matrices. |
| `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). |
| `lowmem` | logical. Should this function be run in a way that saves memory? If TRUE, will run FAR slower but use almost no ram. If FALSE, lots of ram is used but it will run quickly. For example, on an 800x1600 matrix, with nperm= 1000, n_cores=12 and lowmem=F, will use roughly 90 GB ram but finish in one minute. With the same settings but lowmem=T, will use less than 1 GB ram but take over an hour. A progress bar is shown if TRUE (DEFAULT: FALSE). |

`vec_wvar` logical. For vector input, should old weighted-variance approach be used instead of transforming the vector into a euclidean distance matrix? (DEFAULT: FALSE).

`phy_ent` logical. For phylogenetic input, should SES of phylogenetic entropy be used instead of trandforming the phylogeny into patristic distances using tree2mat? (DEFAULT: FALSE).

`denom_type` string. Type of denominator (d) to use for SES. SES is the difference between empirical and mean simulated specificity, divided by d (DEFAULT: global_unif).

**species_sim:** d for species s is calculated as the standard deviation of specificities calculated from permuted abundances of s. This makes SES of specificity counterintuitively sensitive to occupancy, where species with high occupancy have more extreme SES of specificity than rare species, due to their more deterministic sim specificities. Not suggested.

**global_sim:** d is same for all species. Calculated as standard deviation of all specificities calculated from permuted species. SES is much less sensitive to occupancy, and intuitively, species with greater occupancy have less extreme SES of specificity than rare species. However, using this d makes results from different abundance matrices NOT comparable, because d is a function of the distribution of species distributions unique to abunds_mat. Results for different variables and the same abundance matrix are still comparable, though (obviously using same denom_type and other parameters).

**global_unif:** d is same for all species. Calculated as variability in specificity under random uniform distribution (beta 1,1) of species abunds. This d is comparable between different abundance matrices and between different variables. SES results using this denom_type are NOT comparable with results from any other. Due to its stability, this is the default option.

**raw:** d is 1 for all species, so SES is not actually standardized (although the column label in output data will still be "SES"). This option only exists for diagnostic purposes, since units of this metric are incomprehensible.

## Value

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

## 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

## Examples

```
# phylogenetic specificity using endophyte data set
attach(endophyte)
```

```
# only analyze species with occupancy >= 20
m <- occ_threshold(prop_abund(zotutable), 20)
ses_host <- phy_or_env_spec(
    abunds_mat=m,
    hosts=metadata$PlantGenus,
    hosts_phylo=supertree,
    n_cores=12
)

# using vazquez null model from bipartite package as an alternate permutation:
# note that the "creating permuted matrices" step will be slow.
library(bipartite)
ses_host_vaz <- phy_or_env_spec(
    abunds_mat=m,
    hosts=metadata$PlantGenus,
    hosts_phylo=supertree,
    n_cores=12,
    sim_fun=function(m){bipartite::vaznull(1, m)[[1]]},
)

# compare naive permutation vs. vazquez:
plot(ses_host$SES, ses_host_vaz$SES, ylab="bipartite::vaznull", xlab="naive")
abline(h=0);abline(v=0)
hist(ses_host_vaz$SES)
hist(ses_host$SES)

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

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

## 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_specificities *plot_specificities*

---

## Description

Visualizes results from phy_or_env_spec

## Usage

```
plot_specificities(specs_list, n_bins = 20, col_sig = "black",
  col_nsig = "gray", col_bord = NA, alpha = 0.05, label_cex = 0.6)
```

## Arguments

| | |
|---|---|
| specs_list | list of data.frames. Each data.frame must be an output from phy_or_env_spec; must have columns "SES" 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). |

## Value

returns nothing (a plot is made).

## Author(s)

John L. Darcy

## Examples

```
none yet written.
```

---

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])
```

---

`pval_from_perms` *pval_from_perms*

---

## Description

Calculates P-value for permutation tests.

## Usage

```
pval_from_perms(emp, perm, tails, threshold = 100)
```

## Arguments

| | |
|---|---|
| `emp` | Numeric scalar. An empirical test statistic value. |
| `perm` | Numeric vector. Test statistic values similar to emp, but calculated from permuted data. |
| `tails` | integer.<br>**1:** Left tail only.<br>**2:** 2-tailed test.<br>**3:** Right tail only.<br>**0:** No test, P=1. |
| `threshold` | integer. Minimum number n of non-NA values in perm that are acceptable. If n < threshold, P=NA (DEFAULT: 100). |

## Value

a P-value.

## Author(s)

John L. Darcy

## 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

## rao_quad_ent *rao_quad_ent*

### Description

Calculates Rao's (1982) quadratic entropy (FDq) from a distance matrix and a vector of weights (e.g. relative abundance data). In simple terms, FDq is sum product of distances and pairwise products of weights. Default operation in this function is to then divide FDq by the sum of pairwise weights, to give a weighted mean of distances (FDq*). This gives units of distance, and also makes the metric insensitive to the sum of weights, i.e. weights do not need to be normalized before calculation. This behavior can be disabled by setting raw=TRUE, which will give FDq.

### Usage

```
rao_quad_ent(d, w, raw = FALSE)
```

### Arguments

| | |
|---|---|
| `d` | numeric dist. Distances, as a dist object. Note that dist objects can easily be made from square matrices using as.dist(), or euclidean distances can be calculated from numeric vectors using dist(). |
| `w` | numeric vector. Per-sample weights, as a vector. For example, abundances of a species across samples. w MUST be sorted such that it corresponds to rows of as.matrix(d). Thus, w must have length l such that (l^2-l)/2 = length(d). |
| `raw` | logical. If true, FDq will be returned. If false, FDq* will be returned (DEFAULT: FALSE). |

### Value

A single value.

### Author(s)

John L. Darcy

### References

Rao R (1982) Diversity: its measurement, decomposition, apportionment and analysis. Sankhyā: The Indian Journal of Statistics 44(1).

### Examples

```
none yet written
```

---

## `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.

## 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:<br>**Hp:** Phylogenetic Entropy. Insensitive to 0 weights, cannot increase with removal of taxa. Allen et al. 2009.<br>**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.<br>**PD:** Original Faith's Phylogenetic Diversity. Unweighted. Simply a sum of branch- lengths in your tree (but only for taxa in s). Faith 1992. |

## Details

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.

## 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.

## See Also

rao_quad_ent, a phylogenetic diversity measure that uses a distance matrix instead of a phylogenetic tree.

## Examples

```
library(geiger)
set.seed(12345)
s_phylo <- get(data(geospiza))$phy
w <- sample(c(0, 1), replace=T, size=10)
s <- sample(s_phylo$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:<br>**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. Rao R (1982) Diversity: its measurement, decomposition, apportionment and analysis. Sankhyā: The Indian Journal of Statistics 44(1).

### Examples

```
none yet written.
```

# Index