

Package ‘rdiversity’

June 19, 2018

Type Package

Title Measurement and Partitioning of Similarity-Sensitive Biodiversity

Version 1.2.1

Date 2018-06-12

Maintainer Sonia Mitchell <s.mitchell.2@research.gla.ac.uk>

URL <https://github.com/boydorr/rdiversity>

BugReports <https://github.com/boydorr/rdiversity/issues>

Description Provides a framework for the measurement and partitioning of the (similarity-sensitive) biodiversity of a metacommunity and its constituent subcommunities. Richard Reeve, et al. (2016) <arXiv:1404.6520v3>.

License GPL-3

Imports methods, ggplot2, ggthemes, ape, phangorn, plyr, tidyr, tibble, phytools, reshape2

Suggests testthat, knitr, rmarkdown

RoxygenNote 6.0.1

Collate 'chainsaw.R' 'check_partition.R' 'check_phypartition.R' 'check_similarity.R' 'class-metacommunity.R' 'class-powermean.R' 'class-relativeentropy.R' 'metacommunity.R' 'diversity-components.R' 'diversity-measures.R' 'get_title.R' 'hs_parameters.R' 'inndiv.R' 'metadiv.R' 'phy_abundance.R' 'phy_struct.R' 'plot_diversity.R' 'power_mean.R' 'powermean.R' 'qD.R' 'subdiv.R' 'qDZ.R' 'qDZ_single.R' 'qD_single.R' 'rdiversity-package.R' 'relativeentropy.R' 'repartition.R' 'similarity_shimatani.R' 'smatrix.R' 'summarise.R' 'tbar.R' 'zmatrix.R'

Encoding UTF-8

NeedsCompilation no

Author Sonia Mitchell [aut, cre] (<<https://orcid.org/0000-0003-1536-2066>>),
Richard Reeve [aut] (<<https://orcid.org/0000-0003-2589-8091>>)

Repository CRAN

Date/Publication 2018-06-19 14:53:02 UTC

R topics documented:

rdiversity-package	3
chainsaw	4
check_partition	5
check_phypartition	5
check_similarity	6
get_title	6
hs_parameters	7
inndiv	7
metacommunity	9
metacommunity-class	11
metadiv	12
meta_gamma	14
norm_alpha	15
norm_beta	16
norm_meta_alpha	17
norm_meta_beta	18
norm_meta_rho	19
norm_rho	20
norm_sub_alpha	21
norm_sub_beta	22
norm_sub_rho	23
phy_abundance	24
phy_struct	24
plot_diversity	25
powermean	26
powermean-class	27
power_mean	28
qD	29
qDZ	29
qDZ_single	30
qD_single	31
raw_alpha	32
raw_beta	33
raw_gamma	34
raw_meta_alpha	35
raw_meta_beta	36
raw_meta_rho	37
raw_rho	38
raw_sub_alpha	39
raw_sub_beta	40
raw_sub_rho	41
relativeentropy	42

rdiversity-package 3

relativeentropy-class	43
repartition	43
similarity_shimatani	44
smatrix	46
subdiv	47
sub_gamma	48
summarise	49
tbar	50
zmatrix	50

Index 52

rdiversity-package *rdiversity*

Description

rdiversity is an R package based around a framework for measuring and partitioning biodiversity using similarity-sensitive diversity measures. It provides functionality for measuring alpha, beta and gamma diversity of metacommunities (*e.g.* ecosystems) and their constituent subcommunities, where similarity may be defined as taxonomic, phenotypic, genetic, phylogenetic, functional, and so on. It uses the diversity measures described in the arXiv paper, *How to partition diversity*.

Details

- For more information go to our GitHub page; <https://github.com/boydorr/rdiversity>
- Please raise an issue if you find any problems; <https://github.com/boydorr/rdiversity/issues>
- This package is cross-validated against our Julia package; <https://github.com/richardreeve/Diversity.jl>

Author(s)

Sonia Mitchell <s.mitchell.2@research.gla.ac.uk>, Richard Reeve <richard.reeve@glasgow.ac.uk>

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2016. How to partition diversity. (<https://arxiv.org/abs/1404.6520>)

chainsaw	<i>Function to cut phylogeny from present day species to a specified depth.</i>
----------	---

Description

Function to cut phylogeny from present day species to a specified depth.

Usage

```
chainsaw(partition, ps, depth)
```

Arguments

partition	proportional abundance of types in the subcommunity as a fraction of the meta-community as a whole (in the phylogenetic case, this corresponds to the proportional abundance of present day species)
ps	phy_struct() output
depth	proportion of total tree height to be conserved (taken as a proportion from the highest tip). Describes how far back we go in the tree, with 0 marking the date of the most recent tip, and 1 marking the most recent common ancestor. Numbers greater than 1 extend the root of the tree

Value

Returns an object of class phy_struct containing a new structural matrix ('@structure') and the original phylogenetic parameters ('@parameters').

Examples

```
tree <- ape::rtree(n = 5)
tree$tip.label <- paste0("sp", seq_along(tree$tip.label))
partition <- cbind(a = c(1,1,1,0,0), b = c(0,1,0,1,1))
row.names(partition) <- tree$tip.label
partition <- partition / sum(partition)
ps <- phy_struct(tree, partition)

a <- chainsaw(partition, ps, depth = 0.9)
b <- chainsaw(partition, ps, depth = 2)
z <- chainsaw(partition, ps, depth = 0)
m <- chainsaw(partition, ps, depth = 1)
```

check_partition	<i>Check partition matrix</i>
-----------------	-------------------------------

Description

check_partition() is used to validate partition matrices.

Usage

```
check_partition(partition)
```

Arguments

partition	two-dimensional matrix of mode numeric with rows as types, columns as sub-communities, and elements containing relative abundances of types in subcommunities. In the case of phylogenetic metacommunities, these are the relative abundances of terminal taxa.
-----------	---

Value

Returns a two-dimensional matrix of mode numeric. If the partition matrix was valid, this should be identical to that which was input as an argument.

check_phypartition	<i>check_phypartition</i>
--------------------	---------------------------

Description

check_phypartition() is used to validate partition matrices for use with phylogenies.

Usage

```
check_phypartition(tip_labels, partition)
```

Arguments

tip_labels	vector containing elements of class character.
partition	two-dimensional matrix of mode numeric with rows as types, columns as sub-communities, and elements containing relative abundances of types in subcommunities. In the case of phylogenetic metacommunities, these are the relative abundances of terminal taxa.

Value

Returns a two-dimensional matrix of mode numeric. If the partition matrix was valid, this should be identical to that which was input as an argument.

check_similarity	<i>Check similarity matrix</i>
------------------	--------------------------------

Description

check_similarity() is used to validate similarity matrices.

Usage

```
check_similarity(partition, similarity)
```

Arguments

partition	two-dimensional matrix of mode numeric with rows as types, columns as sub-communities, and elements containing relative abundances of types in subcommunities. In the case of phylogenetic metacommunities, these are the relative abundances of terminal taxa.
similarity	two-dimensional matrix of mode numeric; contains pair-wise similarity between types.

Value

Returns a two-dimensional matrix of mode numeric. If the similarity matrix was valid, this should be identical to that which was input as an argument.

get_title	<i>Get title</i>
-----------	------------------

Description

Get title

Usage

```
get_title(community_type, measure, symbol = FALSE)
```

Arguments

community_type	object of class character
measure	object of class character
symbol	(optional) by default, output is e.g. "Metacommunity x"; if symbol is set to TRUE, output will be given as "x"

Examples

```
get_title("subcommunity", "normalised alpha")
get_title("metacommunity", "normalised alpha")
get_title("subcommunity", "normalised alpha", TRUE)
get_title("metacommunity", "normalised alpha", TRUE)
```

hs_parameters	<i>Historical species parameters</i>
---------------	--------------------------------------

Description

Extracts various parameters associated with historical species.

Usage

```
hs_parameters(tree)
```

Arguments

tree object of class phylo.

Value

Returns parameters associated with each historic species.

Examples

```
tree <- ape::rtree(n = 5)
tree$tip.label <- paste0("sp", seq_along(tree$tip.label))
hs_parameters(tree)
```

inddiv	<i>Calculate individual-level diversity</i>
--------	---

Description

Generic function for calculating individual-level diversity.

Usage

```
inndiv(data, qs)

## S4 method for signature 'powermean'
inndiv(data, qs)

## S4 method for signature 'relativeentropy'
inndiv(data, qs)

## S4 method for signature 'metacommunity'
inndiv(data, qs)
```

Arguments

data	matrix of mode numeric; containing diversity components.
qs	vector of mode numeric; parameter of conservatism.

Details

data may be input as three different classes:

- `power_mean`: calculates raw and normalised subcommunity alpha, rho or gamma diversity by taking the powermean of diversity components
- `relativeentropy`: calculates raw or normalised subcommunity beta diversity by taking the relative entropy of diversity components
- `metacommunity`: calculates all subcommunity measures of diversity

Value

Returns a standard output of class `tibble`, with columns:

- `measure`: raw or normalised, alpha, beta, rho, or gamma
- `q`: parameter of conservatism
- `type_level`: "subcommunity"
- `type_name`: label attributed to type
- `partition_level`: level of diversity, *i.e.* subcommunity
- `partition_name`: label attributed to partition
- `diversity`: calculated subcommunity diversity

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2016. How to partition diversity. arXiv 1404.6520v3:1–9.

Examples

```

# Define metacommunity
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate subcommunity gamma diversity (takes the power mean)
g <- raw_gamma(meta)
inndiv(g, 0:2)

# Calculate subcommunity beta diversity (takes the relative entropy)
b <- raw_beta(meta)
inndiv(b, 0:2)

# Calculate all measures of individual diversity
inndiv(meta, 0:2)

```

metacommunity	<i>Coerce to metacommunity</i>
---------------	--------------------------------

Description

Functions to check if an object is a metacommunity or coerce an object into a metacommunity.

Usage

```

metacommunity(partition, similarity, ...)

## S4 method for signature 'data.frame,missing'
metacommunity(partition)

## S4 method for signature 'numeric,missing'
metacommunity(partition)

## S4 method for signature 'matrix,missing'
metacommunity(partition)

## S4 method for signature 'data.frame,matrix'
metacommunity(partition, similarity)

## S4 method for signature 'numeric,matrix'
metacommunity(partition, similarity)

## S4 method for signature 'matrix,matrix'
metacommunity(partition, similarity)

```

```
## S4 method for signature 'missing,phylo'
metacommunity(partition, similarity, depth = 1)

## S4 method for signature 'numeric,phylo'
metacommunity(partition, similarity, depth = 1)

## S4 method for signature 'data.frame,phylo'
metacommunity(partition, similarity, depth = 1)

## S4 method for signature 'matrix,phylo'
metacommunity(partition, similarity, depth = 1)

is.metacommunity(x)
```

Arguments

<code>partition</code>	two-dimensional matrix of mode <code>numeric</code> with rows as types, columns as sub-communities, and elements containing the relative abundances of types in sub-communities. For phylogenetic diversity, see <i>Details</i> .
<code>similarity</code>	(optional) two-dimensional matrix of mode <code>numeric</code> , with rows as types, columns as types, and elements containing the pairwise similarity between types. For phylogenetic diversity, see <i>Details</i> .
<code>...</code>	(optional) additional arguments, especially:
<code>depth</code>	(optional; and for phylogenetic metacommunities only) how much evolutionary history should be retained, with 0 marking the most recent present-day species, and 1 (the default) marking the most recent common ancestor. Numbers greater than 1 extend the root of the tree.
<code>x</code>	any R object

Details

When calculating phylogenetic diversity either:

- set `partition` as the relative abundance of present-day species, with `similarity` as an object of class `phylo`, from which the relative abundance and pairwise similarity of historical species will be calculated; or
- set `partition` as the relative abundance of historical species, with `similarity` as the pairwise similarity of historical species.

Value

Returns an object of class `metacommunity` (see *Fields*).

Returns `TRUE` if its argument is a `metacommunity`, `FALSE` otherwise.

Fields

`type_abundance` two-dimensional matrix of mode `numeric` with rows as types, columns as sub-communities, and elements containing relative abundances of types in subcommunities. In the

phylogenetic case, this corresponds to the proportional abundance of historic species, which is calculated from the proportional abundance of present day species.

`similarity` two-dimensional matrix of mode numeric with rows as types, columns as types, and elements containing pairwise similarities between types

`ordinariness` two-dimensional matrix of mode numeric with rows as types, columns as subcommunities, and elements containing the ordinariness of types within subcommunities

`subcommunity_weights` vector of mode numeric; contains subcommunity weights

`type_weights` two-dimensional matrix of mode numeric, with rows as types, columns as subcommunities, and elements containing weights of types within a subcommunity

`raw_abundance` [Phylogenetic] two-dimensional matrix of mode numeric with rows as types, columns as subcommunities, and elements containing the relative abundance of present day species

`raw_structure` [Phylogenetic] two-dimensional matrix of mode numeric with rows as historical species, columns as present day species, and elements containing historical species lengths within lineages

`parameters` [Phylogenetic] tibble containing parameters associated with each historic species in the phylogeny

See Also

[metacommunity-class](#)

Examples

```
tree <- ape::rtree(n = 5)
tree$tip.label <- paste0("sp", seq_along(tree$tip.label))
partition <- cbind(a = c(1,1,1,0,0), b = c(0,1,0,1,1))
row.names(partition) <- tree$tip.label
partition <- partition / sum(partition)

a <- metacommunity(partition, tree)
b <- metacommunity(partition)
```

`metacommunity-class` *metacommunity-class*

Description

Container for class `metacommunity`.

Usage

```
## S4 method for signature 'metacommunity'
show(object)
```

Arguments

object object of class metacommunity

Fields

type_abundance two-dimensional matrix of mode numeric with rows as types, columns as sub-communities, and elements containing relative abundances of types in subcommunities. In the phylogenetic case, this corresponds to the proportional abundance of historic species, which is calculated from the proportional abundance of present day species.

similarity two-dimensional matrix of mode numeric with rows as types, columns as types, and elements containing pairwise similarities between types

ordinariness two-dimensional matrix of mode numeric with rows as types, columns as sub-communities, and elements containing the ordinariness of types within subcommunities

subcommunity_weights vector of mode numeric; contains subcommunity weights

type_weights two-dimensional matrix of mode numeric, with rows as types, columns as sub-communities, and elements containing weights of types within a subcommunity

raw_abundance [Phylogenetic] two-dimensional matrix of mode numeric with rows as types, columns as subcommunities, and elements containing the relative abundance of present day species

raw_structure [Phylogenetic] two-dimensional matrix of mode numeric with rows as historical species, columns as present day species, and elements containing historical species lengths within lineages

parameters [Phylogenetic] tibble containing parameters associated with each historic species in the phylogeny

metadiv

Calculate metacommunity-level diversity

Description

Generic function for calculating metacommunity-level diversity.

Usage

```
metadiv(data, qs)

## S4 method for signature 'powermean'
metadiv(data, qs)

## S4 method for signature 'relativeentropy'
metadiv(data, qs)

## S4 method for signature 'metacommunity'
metadiv(data, qs)
```

Arguments

data see *Details*
 qs vector of mode numeric; parameter of conservatism.

Details

data may be input as one of three different classes:

- powermean: raw or normalised metacomunity alpha, rho or gamma diversity components; will calculate metacommunity-level raw or normalised metacomunity alpha, rho or gamma diversity
- relativeentropy: raw or normalised metacommunity beta diversity components; will calculate metacommunity-level raw or normalised metacommunity beta diversity
- metacommunity: will calculate all metacommunity measures of diversity

Value

Returns a standard output of class tibble, with columns:

- measure: raw or normalised, alpha, beta, rho, or gamma
- q: parameter of conservatism
- type_level: "metacommunity"
- type_name: label attributed to type
- partition_level: level of diversity, *i.e.* metacommunity
- partition_name: label attributed to partition
- diversity: calculated metacommunity diversity

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2016. How to partition diversity. arXiv 1404.6520v3:1–9.

Examples

```
# Define metacommunity
pop <- data.frame(a = c(1,3), b = c(1,1))
pop <- pop / sum(pop)
meta <- metacommunity(pop)

# Calculate metacommunity gamma diversity (takes the power mean)
g <- raw_gamma(meta)
metadiv(g, 0:2)

# Calculate metacommunity beta diversity (takes the relative entropy)
b <- raw_beta(meta)
metadiv(b, 0:2)

# Calculate all measures of metacommunity diversity
```

```
metadiv(meta, 0:2)
```

meta_gamma	<i>Metacommunity gamma diversity</i>
------------	--------------------------------------

Description

Calculates similarity-sensitive metacommunity gamma diversity (the metacommunity similarity-sensitive diversity). This measure may be calculated for a series of orders, represented as a vector of qs .

Usage

```
meta_gamma(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of q values

Value

Returns a standard output of class tibble, with columns: measure, (raw gamma), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* metacommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate metacommunity gamma diversity
meta_gamma(meta, 0:2)
```

norm_alpha	<i>Normalised alpha (low level diversity component)</i>
------------	---

Description

Calculates the low-level diversity component necessary for calculating normalised alpha diversity.

Usage

```
norm_alpha(meta)
```

Arguments

meta object of class metacommunity.

Details

Values generated from norm_alpha() may be input into subdiv() and metadiv() to calculate normalised subcommunity/metacommunity alpha diversity.

Value

Returns an object of class powermean.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate normalised alpha component
norm_alpha(meta)
```

norm_beta	<i>Normalised beta (low level diversity component)</i>
-----------	--

Description

Calculates the low-level diversity component necessary for calculating normalised beta diversity.

Usage

```
norm_beta(meta)
```

Arguments

meta object of class metacommunity.

Details

Values generated from `norm_beta()` may be input into `subdiv()` and `metadiv()` to calculate normalised subcommunity/metacommunity beta diversity.

Value

Returns an object of class `relativeentropy`.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate normalised beta component
norm_beta(meta)
```

norm_meta_alpha	<i>Normalised metacommunity alpha diversity</i>
-----------------	---

Description

Calculates similarity-sensitive normalised metacommunity alpha diversity (the average similarity-sensitive diversity of subcommunities). This measure may be calculated for a series of orders, represented as a vector of qs .

Usage

```
norm_meta_alpha(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of q values

Value

Returns a standard output of class tibble, with columns: measure, (norm alpha), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* metacommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate normalised metacommunity alpha diversity
norm_meta_alpha(meta, 0:2)
```

norm_meta_beta	<i>Normalised metacommunity beta diversity</i>
----------------	--

Description

Calculates similarity-sensitive normalised metacommunity beta diversity (the effective number of distinct subcommunities). This measure may be calculated for a series of orders, represented as a vector of qs .

Usage

```
norm_meta_beta(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of q values

Value

Returns a standard output of class tibble, with columns: measure, (norm beta), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* metacommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate normalised metacommunity beta diversity
norm_meta_beta(meta, 0:2)
```

norm_meta_rho	<i>Normalised metacommunity rho diversity</i>
---------------	---

Description

Calculates similarity-sensitive normalised metacommunity rho diversity (the average representativeness of subcommunities). This measure may be calculated for a series of orders, represented as a vector of qs.

Usage

```
norm_meta_rho(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of q values

Value

Returns a standard output of class tibble, with columns: measure, (norm rho), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* metacommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate normalised metacommunity rho diversity
norm_meta_rho(meta, 0:2)
```

norm_rho	<i>Normalised rho (low level diversity component)</i>
----------	---

Description

Calculates the low-level diversity component necessary for calculating normalised rho diversity.

Usage

```
norm_rho(meta)
```

Arguments

meta object of class metacommunity.

Details

Values generated from `norm_rho()` may be input into `subdiv()` and `metadiv()` to calculate normalised subcommunity/metacommunity rho diversity.

Value

Returns an object of class powermean.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate normalised rho component
norm_rho(meta)
```

norm_sub_alpha	<i>Normalised subcommunity alpha diversity</i>
----------------	--

Description

Calculates similarity-sensitive normalised subcommunity alpha diversity (the diversity of subcommunity j in isolation. This measure may be calculated for a series of orders, represented as a vector of qs).

Usage

```
norm_sub_alpha(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of q values

Value

Returns a standard output of class tibble, with columns: measure, (norm alpha), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* subcommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate normalised subcommunity alpha diversity
norm_sub_alpha(meta, 0:2)
```

norm_sub_beta	<i>Normalised subcommunity beta diversity</i>
---------------	---

Description

Calculates similarity-sensitive normalised subcommunity beta diversity (an estimate of the effective number of distinct subcommunities). This measure may be calculated for a series of orders, represented as a vector of *qs*.

Usage

```
norm_sub_beta(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of <i>q</i> values

Value

Returns a standard output of class tibble, with columns: measure, (norm beta), *q* (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* subcommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate normalised subcommunity beta diversity
norm_sub_beta(meta, 0:2)
```

norm_sub_rho	<i>Normalised subcommunity rho diversity</i>
--------------	--

Description

Calculates similarity-sensitive normalised subcommunity rho diversity (the representativeness of subcommunity j). This measure may be calculated for a series of orders, represented as a vector of qs .

Usage

```
norm_sub_rho(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of q values

Value

Returns a standard output of class tibble, with columns: measure, (norm rho), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* subcommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate normalised subcommunity rho diversity
norm_sub_rho(meta, 0:2)
```

phy_abundance	<i>Calculate abundance of historical species</i>
---------------	--

Description

Calculates the relative abundance of historical species.

Usage

```
phy_abundance(partition, structure_matrix)
```

Arguments

partition	two-dimensional matrix of mode numeric with rows as types, columns as sub-communities, and elements containing relative abundances of types in subcommunities. In the case of phylogenetic metacommunities, these are the relative abundances of terminal taxa.
structure_matrix	output\$structure of phy_struct().

Examples

```
tree <- ape::rtree(n = 5)
tree$tip.label <- paste0("sp", seq_along(tree$tip.label))
partition <- cbind(a = c(1,1,1,0,0), b = c(0,1,0,1,1))
row.names(partition) <- tree$tip.label
partition <- partition / sum(partition)
ps <- phy_struct(tree, partition)
structure_matrix <- ps$structure
phy_abundance(partition, structure_matrix)
```

phy_struct	<i>Calculate phylogenetic structure matrix</i>
------------	--

Description

Converts an object into class phylo into class phy_struct.

Usage

```
phy_struct(tree, partition)
```


Arguments

tree	object of class phylo
partition	two-dimensional matrix of mode numeric with rows as types, columns as sub-communities, and elements containing relative abundances of types in subcommunities. In the case of phylogenetic metacommunities, these are the relative abundances of terminal taxa.

Value

Returns a list containing:

\$structure	- each row denotes historical species, columns denote terminal taxa, and elements contain 'branch lengths / t
\$parameters	- information associated with each historical species
\$tree	- object of class phylo

Examples

```
tree <- ape::rtree(n = 5)
tree$tip.label <- paste0("sp", seq_along(tree$tip.label))
partition <- cbind(a = c(1,1,1,0,0), b = c(0,1,0,1,1))
row.names(partition) <- tree$tip.label
partition <- partition / sum(partition)

res <- phy_struct(tree, partition)
```

plot_diversity	<i>Plot diversity</i>
----------------	-----------------------

Description

Simple function to plot diversity profiles.

Usage

```
plot_diversity(data)
```

Arguments

data	object of class tibble (or data.frame); output of subdiv, metadiv, or any of the specific subcommunity- or metacommunity-level diversity functions.
------	---

Examples

```

# Define metacommunity
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate subcommunity beta diversity (takes the relative entropy)
b <- raw_beta(meta)
res <- subdiv(b, 0:2)
plot_diversity(res)

# Calculate all measures of subcommunity diversity
res <- subdiv(meta, 0:2)
plot_diversity(res)

# Try a single population
pop <- c(1,3,4)
meta <- metacommunity(pop)
res <- meta_gamma(meta, 0:2)
plot_diversity(res)

# Calculate all measures of metacommunity diversity
res <- metadiv(meta, 0:2)
plot_diversity(res)

```

powermean

Calculate power mean

Description

Functions to check if an object is a powermean, or coerce an object into a powermean; for `raw_alpha()`, `norm_alpha()`, `raw_rho()`, `norm_rho()`, or `raw_gamma()`.

Usage

```

powermean(results, meta, tag)

is.powermean(x)

## S4 method for signature 'powermean'
show(object)

```

Arguments

`results` matrix of mode numeric; contains values calculated from diversity-term functions `norm_alpha()`, `raw_alpha()`, `raw_rho()`, `norm_rho()`, and `raw_gamma()`

meta	object of class metacommunity; contains proportional abundance of types, pairwise similarity, and other associated variables.
tag	measure
x	any R object
object	object of class powermean

Value

powermean(x) returns an object of class powermean.

is.powermean(x) returns TRUE if its argument is a powermean, FALSE otherwise.

print(x) prints an object object of class powermean

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate subcommunity raw alpha diversity (takes the powermean)
a <- raw_alpha(meta)
class(a)
```

powermean-class	<i>powermean-class</i>
-----------------	------------------------

Description

Container for class powermean.

Fields

output object of class tibble, with columns: measure, (raw alpha, norm alpha, raw rho, etc.), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* subcommunity), partition_name (label attributed to partition), and diversity

results matrix of mode numeric; contains values calculated from diversity-term values output from raw_alpha(), norm_alpha(), raw_rho(), norm_rho(), or raw_gamma()

type_abundance two-dimensional matrix of mode numeric; contains proportional abundance of *types* in the subcommunity as a fraction of the metacommunity as a whole (in the phylogenetic case, this corresponds to the proportional abundance of historic species, which is calculated from the proportional abundance of present day species)

ordinariness two-dimensional matrix of mode numeric; contains ordinariness of types

subcommunity_weights vector of mode numeric; contains subcommunity weights

type_weights two-dimensional matrix of mode numeric; contains weight of types within a subcommunity

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

power_mean	<i>Power mean of vector elements</i>
------------	--------------------------------------

Description

power_mean() calculates the power mean of a set of values.

Usage

```
power_mean(values, order = 1, weights = rep(1, length(values)))
```

Arguments

values	Values for which to calculate mean.
order	Order of power mean.
weights	Weights of elements, normalised to 1 inside function.

Details

Calculates the order-th power mean of a single set of non-negative values, weighted by weights; by default, weights are equal and order is 1, so this is just the arithmetic mean. Equal weights and a order of 0 gives the geometric mean, and an order of -1 gives the harmonic mean.

Value

Weighted power mean

Examples

```
values <- sample(1:50, 5)
power_mean(values)
```

qD	<i>Hill number / naive diversity with no similarity</i>
----	---

Description

Calculates the diversity of a series of columns representing independent populations, for a series of orders, represented as a vector of qs .

Usage

```
qD(populations, qs)
```

Arguments

populations - population counts or proportions.
 qs - vector of values of parameter q .

Details

qD is used to calculate the diversity of a population (in the naive-type case).

Value

Returns an object of class `data.frame` with columns as populations, rows as values of q , and elements containing diversities.

qDZ	<i>Similarity-sensitive diversity</i>
-----	---------------------------------------

Description

Calculates the similarity-sensitive diversity of a series of columns representing independent populations, for a series of orders represented as a vector of qs .

Usage

```
qDZ(meta, qs)
```

Arguments

meta object of class `metacommunity`.
 qs vector of q values.

Value

Returns an object of class `matrix`, with rows as qs columns as subcommunities, and elements containing diversity values.

Examples

```
pop <- sample(1:50, 5)

# Create similarity matrix
Z <- diag(1, length(pop))
Z[Z==0] <- 0.4

dat <- metacommunity(pop, Z)

# Calculate similarity-sensitive diversity of order 0 (species richness)
qDZ(dat, 0)
```

qDZ_single

Similarity-sensitive diversity of a single population

Description

Calculates the similarity-sensitive diversity of order q of a single population with given relative proportions.

Usage

```
qDZ_single(proportions, q, Z = diag(nrow(proportions)), Zp = Z %*%
  proportions)
```

Arguments

proportions	vector of mode <code>numeric</code> ; contains the relative proportions of different individuals/types in a population.
q	object of class <code>numeric</code> ; contains the order of diversity measurement.
Z	two-dimensional matrix of mode <code>numeric</code> ; contains the pair-wise similarity of individuals/types in a population.
Zp	two-dimensional matrix of mode <code>numeric</code> ; contains the ordinariness of individuals/types in population.

Value

Returns the similarity-sensitive diversity of order q .

Examples

```
pop <- sample(1:50, 5)

# Create similarity matrix
Z <- diag(1, length(pop))
Z[Z==0] <- 0.4

# Calculate similarity-sensitive diversity of order 0 (species richness)
qDZ_single(pop, 0, Z)
```

qD_single

Hill number / naive diversity of a single population

Description

Calculates the Hill number (naive diversity) of order q of a single population with given relative proportions.

Usage

```
qD_single(proportions, q)
```

Arguments

proportions vector of mode `numeric`; contains the relative proportions of different individuals/types in a population.

q object of class `numeric`; contains the order of diversity measurement.

Value

Returns the naive diversity of order q .

Examples

```
pop <- sample(1:50, 5)
qD_single(pop, 0)
```

raw_alpha	<i>Raw alpha (low level diversity component)</i>
-----------	--

Description

Calculates the low-level diversity component necessary for calculating alpha diversity.

Usage

```
raw_alpha(meta)
```

Arguments

meta object of class metacommunity.

Details

Values generated from `raw_alpha()` may be input into `subdiv()` and `metadiv()` to calculate raw subcommunity/metacommunity alpha diversity.

Value

Returns an object of class `powermean`.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2016. How to partition diversity. arXiv 1404.6520v3:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate raw alpha component
raw_alpha(meta)
```

raw_beta	<i>Raw beta (low level diversity component)</i>
----------	---

Description

Calculates the low-level diversity component necessary for calculating raw beta diversity.

Usage

```
raw_beta(meta)
```

Arguments

meta object of class metacommunity.

Details

Values generated from `raw_beta()` may be input into `subdiv()` and `metadiv()` to calculate raw subcommunity/metacommunity beta diversity.

Value

Returns an object of class `relativeentropy`.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate raw beta component
raw_beta(meta)
```

raw_gamma	<i>Gamma (low level diversity component)</i>
-----------	--

Description

Calculates the low-level diversity component necessary for calculating gamma diversity.

Usage

```
raw_gamma(meta)
```

Arguments

meta object of class metacommunity.

Details

Values generated from `raw_gamma()` may be input into `subdiv()` and `metadiv()` to calculate sub-community/metacommunity gamma diversity.

Value

Returns an object of class powermean.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate gamma component
raw_gamma(meta)
```

raw_meta_alpha	<i>Raw metacommunity alpha diversity</i>
----------------	--

Description

Calculates similarity-sensitive raw metacommunity alpha diversity (the naive-community metacommunity diversity). This measure may be calculated for a series of orders, represented as a vector of qs .

Usage

```
raw_meta_alpha(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of q values

Value

Returns a standard output of class tibble, with columns: measure, (raw alpha), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* metacommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate raw metacommunity alpha diversity
raw_meta_alpha(meta, 0:2)
```

raw_meta_beta	<i>Raw metacommunity beta diversity</i>
---------------	---

Description

Calculates similarity-sensitive raw metacommunity beta diversity (the average distinctiveness of subcommunities). This measure may be calculated for a series of orders, represented as a vector of *qs*.

Usage

```
raw_meta_beta(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of <i>q</i> values

Value

Returns a standard output of class tibble, with columns: measure, (raw beta), *q* (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* metacommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate raw metacommunity beta diversity
raw_meta_beta(meta, 0:2)
```

raw_meta_rho	<i>Raw metacommunity rho diversity</i>
--------------	--

Description

Calculates similarity-sensitive raw metacommunity rho diversity (the average redundancy of sub-communities). This measure may be calculated for a series of orders, represented as a vector of qs.

Usage

```
raw_meta_rho(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of q values

Value

Returns a standard output of class tibble, with columns: measure, (raw rho), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* metacommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate metacommunity rho diversity
raw_meta_rho(meta, 0:2)
```

raw_rho	<i>Raw rho (low level diversity component)</i>
---------	--

Description

Calculates the low-level diversity component necessary for calculating raw rho diversity.

Usage

```
raw_rho(meta)
```

Arguments

meta object of class metacommunity.

Details

Values generated from `raw_rho()` may be input into `subdiv()` and `metadiv()` to calculate raw subcommunity/metacommunity rho diversity.

Value

Returns an object of class powermean.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate raw rho component
raw_rho(meta)
```

raw_sub_alpha	<i>Raw subcommunity alpha diversity</i>
---------------	---

Description

Calculates similarity sensitive raw subcommunity alpha diversity (an estimate of naive-community metacommunity diversity). This measure may be calculated for a series of orders, represented as a vector of qs.

Usage

```
raw_sub_alpha(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of q values

Value

Returns a standard output of class tibble, with columns: measure, (raw alpha), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* subcommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2016. How to partition diversity. arXiv 1404.6520v3:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate raw subcommunity alpha diversity
raw_sub_alpha(meta, 0:2)
```

raw_sub_beta	<i>Raw subcommunity beta diversity</i>
--------------	--

Description

Calculates similarity-sensitive raw subcommunity beta diversity (the distinctiveness of subcommunity j). This measure may be calculated for a series of orders, represented as a vector of qs .

Usage

```
raw_sub_beta(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of q values

Value

Returns a standard output of class tibble, with columns: measure, (raw beta), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* subcommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate raw subcommunity beta diversity
raw_sub_beta(meta, 0:2)
```

raw_sub_rho	<i>Raw subcommunity rho diversity</i>
-------------	---------------------------------------

Description

Calculates similarity-sensitive raw subcommunity rho diversity (the redundancy of subcommunity j). This measure may be calculated for a series of orders, represented as a vector of qs .

Usage

```
raw_sub_rho(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of q values

Value

Returns a standard output of class tibble, with columns: measure, (raw rho), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* subcommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate raw subcommunity rho diversity
raw_sub_rho(meta, 0:2)
```

relativeentropy *Calculate relative entropy*

Description

Functions to check if an object is a relativeentropy, or coerce an object into a relativeentropy; for raw_beta() or norm_beta().

Usage

```
relativeentropy(results, meta, tag)

is.relativeentropy(x)

## S4 method for signature 'relativeentropy'
show(object)
```

Arguments

results	matrix of mode numeric; contains values calculated from diversity-term functions raw_beta() and norm_beta()
meta	object of class metacommunity
tag	measure
x	any R object
object	object of class relativeentropy

Value

object of class relativeentropy
is.relativeentropy(x) returns TRUE if its argument is a relativeentropy, FALSE otherwise.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate raw subcommunity beta diversity
a <- raw_beta(meta)
class(a)
```

relativeentropy-class *relativeentropy-class*

Description

Container for class `relativeentropy`.

Fields

`results` object of class `matrix` of mode `numeric`; contains diversity term values output from `raw_beta()` or `norm_beta()`

`measure` `measure`

`type_abundance` two-dimensional `matrix` of mode `numeric`; contains proportional abundance of *types* in the subcommunity as a fraction of the metacommunity as a whole (in the phylogenetic case, this corresponds to the proportional abundance of historic species, which is calculated from the proportional abundance of present day species)

`ordinariness` two-dimensional `matrix` of mode `numeric`; contains ordinariness of types

`subcommunity_weights` vector of mode `numeric`; contains subcommunity weights

`type_weights` two-dimensional `matrix` of mode `numeric`; contains weight of types within a subcommunity

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. *arXiv 1404.6520:1–9*.

`repartition` *Repartition metacommunity*

Description

Randomly reshuffles the relative abundance of types (*e.g.* species) in a metacommunity (whilst maintaining the relationship between the relative abundance of a particular species across subcommunities). In the case of a phylogenetic metacommunity, the relative abundance of terminal taxa are randomly reshuffled and the relative abundance of types (historical species) are calculated from the resulting partition.

Usage

`repartition(meta, new_partition)`

Arguments

`meta` object of class `metacommunity`.

`new_partition` proportional abundance of *types* in the subcommunity as a fraction of the metacommunity as a whole (in the phylogenetic case, this corresponds to the proportional abundance of terminal taxa). If this argument is missing, all species/tips will be shuffled.

Value

Returns an object of class `metacommunity`.

Examples

```
tree <- ape::rtree(n = 5)
tree$tip.label <- paste0("sp", seq_along(tree$tip.label))
partition <- cbind(a = sample(5,5), b = sample(5,5))
row.names(partition) <- tree$tip.label
partition <- partition / sum(partition)
meta <- metacommunity(partition, tree)
meta@raw_abundance

a <- repartition(meta)
a@raw_abundance

# Non-phylogenetic example
meta <- metacommunity(partition)
meta@type_abundance
a <- repartition(meta)
a@type_abundance
```

similarity_shimatani *Taxonomic similarity matrix*

Description

Calculates taxonomic similarity based on Shimatani's index of taxonomic similarity (see *Details*).

Usage

```
similarity_shimatani(data, lookup)
```

Arguments

`data` $S * N$ matrix; population counts

`lookup` `data.frame` with `colnames = c('Species', 'Genus', 'Family', 'Subclass')`

Details

Shimatani's taxonomic similarity index is defined:

```

species.similarity  1
genus.similarity    0.75
family.similarity   0.5
subclass.similarity 0.25
other.similarity     0

```

Value

Returns an $S * S$ matrix; pair-wise taxonomic similarity

References

Shimatani, K. 2001. On the measurement of species diversity incorporating species differences. *Oikos* 93:135–147.

Examples

```

pop <- sample(1:50, 4)
lookup <- data.frame(Subclass=c("Sapindales", "Malvales", "Fabales",
                                "Fabales"),
                    Family=c("Burseraceae", "Bombacaceae", "Fabaceae",
                              "Fabaceae"),
                    Genus=c("Protium", "Quararibea", "Swartzia",
                             "Swartzia"),
                    Species= c("tenuifolium", "asterolepis",
                                "simplex var.grandiflora",
                                "simplex var.ochnacea"))
similarity <- similarity_shimatani(pop, lookup)

```

smatrix

Phylogenetic similarity matrix (ultrametric)

Description

Function to calculate an ultrametric-similarity matrix.

Usage

```
smatrix(ps)
```

Arguments

ps phy_struct() output.

Value

Returns an *hSxhS* matrix; pair-wise similarity of historic species.

Examples

```

tree <- ape::rtree(n = 5)
tree$tip.label <- paste0("sp", seq_along(tree$tip.label))
partition <- cbind(a = c(1,1,1,0,0), b = c(0,1,0,1,1))
row.names(partition) <- tree$tip.label
partition <- partition / sum(partition)

ps <- phy_struct(tree, partition)
smatrix(ps)

```

subdiv	<i>Calculate subcommunity-level diversity</i>
--------	---

Description

Generic function for calculating subcommunity-level diversity.

Usage

```

subdiv(data, qs)

## S4 method for signature 'powermean'
subdiv(data, qs)

## S4 method for signature 'relativeentropy'
subdiv(data, qs)

## S4 method for signature 'metacommunity'
subdiv(data, qs)

```

Arguments

data	matrix of mode numeric; containing diversity components.
qs	vector of mode numeric; parameter of conservatism.

Details

data may be input as one of three different classes:

- powermean: raw or normalised metacomunity alpha, rho or gamma diversity components; will calculate subcommunity-level raw or normalised metacomunity alpha, rho or gamma diversity
- relativeentropy: raw or normalised metacomunity beta diversity components; will calculate subcommunity-level raw or normalised metacomunity beta diversity
- metacommunity: will calculate all subcommunity measures of diversity

Value

Returns a standard output of class `tibble`, with columns:

- `measure`: raw or normalised, alpha, beta, rho, or gamma
- `q`: parameter of conservatism
- `type_level`: "subcommunity"
- `type_name`: label attributed to type
- `partition_level`: level of diversity, *i.e.* subcommunity
- `partition_name`: label attributed to partition
- `diversity`: calculated subcommunity diversity

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2016. How to partition diversity. arXiv 1404.6520v3:1–9.

Examples

```
# Define metacommunity
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate subcommunity gamma diversity (takes the power mean)
g <- raw_gamma(meta)
subdiv(g, 0:2)

# Calculate subcommunity beta diversity (takes the relative entropy)
b <- raw_beta(meta)
subdiv(b, 0:2)

# Calculate all measures of subcommunity diversity
subdiv(meta, 0:2)
```

sub_gamma

Subcommunity gamma diversity

Description

Calculates similarity-sensitive subcommunity gamma diversity (the contribution per individual toward metacommunity diversity). This measure may be calculated for a series of orders, represented as a vector of `qs`.

Usage

```
sub_gamma(meta, qs)
```

Arguments

meta	object of class metacommunity
qs	vector of q values

Value

Returns a standard output of class tibble, with columns: measure, (raw gamma), q (parameter of conservatism), type_level (), type_name (label attributed to type), partition_level (level of diversity, *i.e.* subcommunity), partition_name (label attributed to partition), and diversity.

References

Reeve, R., T. Leinster, C. Cobbold, J. Thompson, N. Brummitt, S. Mitchell, and L. Matthews. 2014. How to partition diversity. arXiv 1404.6520:1–9.

Examples

```
pop <- data.frame(a = c(1,3), b = c(1,1))
row.names(pop) <- paste0("sp", 1:2)
pop <- pop/sum(pop)
meta <- metacommunity(pop)

# Calculate subcommunity gamma diversity
sub_gamma(meta, 0:2)
```

 summarise

Summary function

Description

This function converts columns of an array (each representing population counts) into proportions, so that each column sums to 1.

Usage

```
summarise(populations, normalise = TRUE)
```

Arguments

populations	An $S \times N$ array whose columns are counts of individuals.
normalise	Normalise probability distribution to sum to 1 for each column rather than just along each set.

Value

Returns an array whose columns are proportions.

tbar	<i>Calculate T_bar</i>
------	------------------------

Description

Function to calculate T_bar.

Usage

```
tbar(partition, structure_matrix)
```

Arguments

partition two-dimensional matrix of mode numeric with rows as types, columns as sub-communities, and elements containing relative abundances of types in subcommunities. In the case of phylogenetic metacommunities, these are the relative abundances of terminal taxa.

structure_matrix output\$structure of phy_struct(); each row denotes historic species, columns denote terminal taxa, and elements contain branch lengths.

zmatrix	<i>Similarity matrix</i>
---------	--------------------------

Description

Function to calculate a phylogenetic similarity matrix.

Usage

```
zmatrix(partition, s, ps)
```

Arguments

partition two-dimensional matrix of mode numeric with rows as types, columns as sub-communities, and elements containing relative abundances of types in subcommunities. In the case of phylogenetic metacommunities, these are the relative abundances of terminal taxa.

s smatrix() output; ultrametric-similarity matrix.

ps phy_struct() output.

Value

Returns an *hSxhS* matrix; pair-wise similarity of historic species.

Examples

```
tree <- ape::rtree(n = 5)
tree$tip.label <- paste0("sp", seq_along(tree$tip.label))
partition <- cbind(a = c(1,1,1,0,0), b = c(0,1,0,1,1))
row.names(partition) <- tree$tip.label
partition <- partition / sum(partition)
ps <- phy_struct(tree, partition)
s <- smatrix(ps)

zmatrix(partition, s, ps)
```

Index

chainsaw, 4
check_partition, 5
check_phypartition, 5
check_similarity, 6

get_title, 6

hs_parameters, 7

inndiv, 7
inndiv,metacommunity-method (inndiv), 7
inndiv,powermean-method (inndiv), 7
inndiv,relativeentropy-method (inndiv), 7
is.metacommunity (metacommunity), 9
is.powermean (powermean), 26
is.relativeentropy (relativeentropy), 42

meta_gamma, 14
metacommunity, 9
metacommunity,data.frame,matrix-method (metacommunity), 9
metacommunity,data.frame,missing-method (metacommunity), 9
metacommunity,data.frame,phylo-method (metacommunity), 9
metacommunity,data.frame-method (metacommunity), 9
metacommunity,data.frame-method,matrix-method (metacommunity), 9
metacommunity,data.frame-method,phylo-method (metacommunity), 9
metacommunity,matrix,matrix-method (metacommunity), 9
metacommunity,matrix,missing-method (metacommunity), 9
metacommunity,matrix,phylo-method (metacommunity), 9
metacommunity,matrix-method (metacommunity), 9
metacommunity,matrix-method,phylo-method (metacommunity), 9
metacommunity,missing,phylo-method (metacommunity), 9
metacommunity,numeric,matrix-method (metacommunity), 9
metacommunity,numeric,missing-method (metacommunity), 9
metacommunity,numeric,phylo-method (metacommunity), 9
metacommunity,numeric-method (metacommunity), 9
metacommunity,numeric-method,matrix-method (metacommunity), 9
metacommunity,numeric-method,phylo-method (metacommunity), 9
metacommunity,phylo-method (metacommunity), 9
metacommunity-class, 11
metadiv, 12
metadiv,metacommunity-method (metadiv), 12
metadiv,powermean-method (metadiv), 12
metadiv,relativeentropy-method (metadiv), 12

norm_alpha, 15
norm_beta, 16
norm_meta_alpha, 17
norm_meta_beta, 18
norm_meta_rho, 19
norm_rho, 20
norm_sub_alpha, 21
norm_sub_beta, 22
norm_sub_rho, 23

phy_abundance, 24
phy_struct, 24
plot_diversity, 25
power_mean, 28

powermean, 26
powermean-class, 27

qD, 29
qD_single, 31
qDZ, 29
qDZ_single, 30

raw_alpha, 32
raw_beta, 33
raw_gamma, 34
raw_meta_alpha, 35
raw_meta_beta, 36
raw_meta_rho, 37
raw_rho, 38
raw_sub_alpha, 39
raw_sub_beta, 40
raw_sub_rho, 41

rdiversity (rdiversity-package), 3
rdiversity-package, 3
relativeentropy, 42
relativeentropy-class, 43
repartition, 43

show,metacommunity-method
 (metacommunity-class), 11
show,powermean-method (powermean), 26
show,relativeentropy-method
 (relativeentropy), 42
similarity_shimatani, 44
smatrix, 46
sub_gamma, 48
subdiv, 47
subdiv,metacommunity-method (subdiv), 47
subdiv,powermean-method (subdiv), 47
subdiv,relativeentropy-method (subdiv),
 47
summarise, 49

tbar, 50

zmatrix, 50