Package: abdiv (via r-universe)

September 12, 2024

Title Alpha and Beta Diversity Measures

Version 0.2.0.9000

Description A collection of measures for measuring ecological diversity. Ecological diversity comes in two flavors: alpha diversity measures the diversity within a single site or sample, and beta diversity measures the diversity across two sites or samples. This package overlaps considerably with other R packages such as 'vegan', 'gUniFrac', 'betapart', and 'fossil'. We also include a wide range of functions that are implemented in software outside the R ecosystem, such as 'scipy', 'Mothur', and 'scikit-bio'. The implementations here are designed to be basic and clear to the reader.

URL https://github.com/kylebittinger/abdiv

BugReports https://github.com/kylebittinger/abdiv/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Suggests testthat (>= 2.1.0), vegan

Imports ape

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

RemoteUrl https://github.com/kylebittinger/abdiv

RemoteRef HEAD

RemoteSha f1e5d38eac073aa36baa6e7832e2e602d83f5686

Contents

abundance_jaccard 2
berger_parker_d 4
binomial_deviance 4

bray_curtis . . . 6
bray_curtis_components . . . 7
canberra . . . 8
chebyshev . . . 9
correlation_distance . . . 10
diversity_measures . . . 11
euclidean . . . 12
faith_pd . . . 14
faith_tree . . . 15
hamming . . . 15
jaccard . . . 16
jaccard_components . . . 19
kempton_taylor_q . . . 19
kullback_leibler_divergence . . . 20
leprieur_tree . . . 21
lozupone_tree . . . 22
manhattan . . . 22
margalef . . . 24
match_to_tree . . . 25
mcintosh_d . . . 26
mcintosh_e . . . 27
menhinick . . . 28
minkowski . . . 28
morisita . . . 29
richness . . . 30
ruzicka . . . 31
shannon . . . 32
simpson . . . 34
strong . . . 35
unifrac . . . 36
unifrac_components . . . 38
weighted_kulczynski_second . . . 40

Index **41**

`abundance_jaccard` *Chao's abundance-weighted indices*

Description

These indices were first developed and introduced as probabilistic, abundance-weighted versions of the Jaccard and Sorenson indices. The abundance information is summarized by two quantities, named U and V in the literature. Each quantity indicates the total abundance of species that are common to both samples: U is the total abundance of such species in the first sample, and V the total in the second sample.

Usage

```
abundance_jaccard(x, y)

abundance_sorenson(x, y)
```

Arguments

x, y	Numeric vectors

Details

In addition to defining the quantities U and V, the articles provide formulas for estimating the value U and V if more sampling was to be performed. In other software packages where these indices are implemented, it is the estimated quantities, U_{est} and V_{est} that are used to calculate the distance. For example, the `chaodist` function in Vegan uses the estimators. In this package, we use the observed values of U and V, rather than the estimated values. Thus, the distances returned here may differ from other software packages that use the estimators rather than the observed values of U and V. In general, the estimators increase U and V, resulting in greater estimated abundance of shared species and thus a smaller distance between samples.

The *abundance-weighted Jaccard* distance is $1 - UV/(U + V - UV)$. Relation of `abundance_jaccard()` to other definitions:

- Differs `vegdist()` with `method = "chao"`. We use the observed values, not the estimators.
- Differs from the `jabund` calculator in Mothur. We use the observed values, not the estimators.

The *abundance-weighted Sorenson* distance is $1 - 2UV/(U + V)$. Relation of `abundance_sorenson()` to other definitions:

- Differs from the `sabund` calculator in Mothur. We use the observed values, not the estimators.

Value

The dissimilarity between `x` and `y`

References

Chao A, Chazdon RL, Colwell R, Shen TJ. A new statistical approach for assessing similarity of species composition with incidence and abundance data. Ecology Letters 2005;8:148-159.

Examples

```
x <- c(5, 2, 3) # u = 0.5
y <- c(0, 5, 5) # v = 1
abundance_jaccard(x, y) # should be 1 - 0.5 / (1 + 0.5 - 0.5) = 0.5
abundance_sorenson(x, y) # should be 1 - 2 * 0.5 / (1 + 0.5) = 0.3333
```

`berger_parker_d` *Berger-Parker dominance*

Description

The Berger-Parker dominance is the proportion of the most abundant species.

Usage

```
berger_parker_d(x)
```

Arguments

`x`	A numeric vector of species counts or proportions.

Details

- Equivalent to `berger_parker_d()` in `skbio.diversity.alpha`.
- Equivalent to the `bergerparker` calculator in Mothur.

Value

The Berger-Parker dominance, $0 < D_{BP} \leq 1$. If the vector sums to zero, the Berger-Parker dominance is undefined, and we return `NaN`.

References

Berger WH, Parker FL. Diversity of Planktonic Foraminifera in Deep-Sea Sediments. Science. 1970;168(3937):1345-1347.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
berger_parker_d(x) # 15 / 28
```

`binomial_deviance` *Binomial deviance and CY index of dissimilarity*

Description

The binomial deviance dissimilarity and the CY (or Cao) index of dissimilarity were created to compare species counts at sites with moderate to large differences.

Usage

```
binomial_deviance(x, y)

cy_dissimilarity(x, y, base = 10, min_value = 0.1)
```

Arguments

`x, y`	Numeric vectors
`base`	Base of the logarithm
`min_value`	Replacement for zero or near-zero values. Values less than `min_value` are replaced with `min_value`.

Details

Both of these measures were designed to be used with whole-numbered counts, and may not make sense for comparing normalized vectors or vectors of species proportions.

For two vectors `x` and `y`, the binomial deviance dissimilarity is

$$d(x,y) = \sum_i \frac{1}{n_i} \left(x_i \log \frac{x_i}{n_i} + y_i \log \frac{y_i}{n_i} - (x_i + y_i) log 2 \right),$$

where $n_i = x_i + y_i$. This value is the weighted average of the deviance for each species, under a binomial model where the expected counts are $n_i/2$ at each site. It was proposed by Anderson and Millar in 2004. Relation to other definitions:

- Equivalent to vegdist() with method = "binomial".

The CY index was proposed by Cao, Williams, and Bark in 1997. For two vectors `x` and `y`, the CY index is

$$d(x,y) = \frac{1}{N} \sum_i \left(\frac{(x_i + y_i) \log_{10}(\frac{x_i + y_i}{2}) - x_i \log_{10}(y_i) - y_i \log_{10}(x_i)}{x_i + y_i} \right),$$

where N is the total number of species in vectors x and y. Double zeros are not considered in the measure.

When either x_i or y_i are zero, they need to be replaced by another value in the CY index to avoid infinities. Cao suggested replacing zero values with 0.1, which is one log lower than the minimum value for whole-numbered counts. Here, we use a `min_value` argument to allow the user set a lower limit on the values. For vectors of species counts, this function follows the formulation of Cao by default.

Relation of the CY index to other definitions:

- Equivalent to the `vegdist()` function with `method = "cao"`, if `base = exp(1)`.

Value

The Binomial deviance or CY index of dissimilarity. The CY index is undefined if all elements of `x` and `y` are zero, in which case we return `NaN`.

References

Anderson MJ, Millar RB. Spatial variation and effects of habitat on temperate reef fish assemblages in northeastern New Zealand. Journal of Experimental Marine Biology and Ecology 2004;305:191–221.

Cao Y, Williams WP, Bark AW. Similarity measure bias in river benthic Aufwuchs community analysis. Water Environment Research 1997;69(1):95-106.

`bray_curtis` *Bray-Curtis distance*

Description

The Bray-Curtis distance is the Manhattan distance divided by the sum of both vectors.

Usage

```
bray_curtis(x, y)
```

Arguments

`x, y`	Numeric vectors

Details

For two vectors `x` and `y`, the Bray-Curtis distance is defined as

$$d(x, y) = \frac{\sum_i |x_i - y_i|}{\sum_i x_i + y_i}.$$

The Bray-Curtis distance is connected to many other distance measures in this package; we try to list some of the more important connections here. Relation to other definitions:

- Equivalent to `vegdist()` with `method = "bray"`.
- Equivalent to the `braycurtis()` function in `scipy.spatial.distance` for positive vectors. They take the absolute value of $x_i + y_i$ in the denominator.
- Equivalent to the `braycurtis` and `odum` calculators in Mothur.
- Equivalent to $D_{14} = 1 - S_{17}$ in Legendre & Legendre.
- The Bray-Curtis distance on proportions is equal to half the Manhattan distance.
- The Bray-Curtis distance on presence/absence vectors is equal to the Sorenson index of dissimilarity.

Value

The Bray-Curtis distance between `x` and `y`. The Bray-Curtis distance is undefined if the sum of all elements in `x` and `y` is zero, in which case we return `NaN`.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
y <- c(10, 2, 0, 1, 1, 0)
bray_curtis(x, y)

# For proportions, equal to half the Manhattan distance
bray_curtis(x / sum(x), y / sum(y))
manhattan(x / sum(x), y / sum(y)) / 2
```

---

bray_curtis_components *Balanced variation and abundance gradient components for abundance data*

Description

Balanced variation and abundance gradient components for abundance data

Usage

```
bray_curtis_balanced(x, y)

bray_curtis_gradient(x, y)

ruzicka_balanced(x, y)

ruzicka_gradient(x, y)
```

Arguments

x, y	Numeric vectors

Value

The balanced variation or abundance gradient component of distance between x and y. This quantity is undefined when either x or y have all elements equal to zero, in which case we return NaN.

References

Baselga A. Separating the two components of abundance-based dissimilarity: balanced changes in abundance vs.abundance gradients. Methods in Ecology and Evolution. 2013;4:552–557.

Baselga A. Partitioning abundance-based multiple-site dissimilarity into components: balanced variation in abundance and abundance gradients. Methods in Ecology and Evolution. 2017;8:799–808.

canberra *Canberra and related distances*

Description

The Canberra distance and Clark's coefficient of divergence are measures that use the absolute difference over the sum for each element of the vectors.

Usage

```
canberra(x, y)

clark_coefficient_of_divergence(x, y)
```

Arguments

`x, y`	Numeric vectors

Details

For vectors `x` and `y`, the Canberra distance is defined as

$$d(x, y) = \sum_i \frac{|x_i - y_i|}{x_i + y_i}.$$

Elements where $x_i + y_i = 0$ are not included in the sum. Relation of `canberra()` to other definitions:

- Equivalent to R's built-in `dist()` function with `method = "canberra"`.
- Equivalent to the `vegdist()` function with `method = "canberra"`, multiplied by the number of entries where `x > 0`, `y > 0`, or both.
- Equivalent to the `canberra()` function in `scipy.spatial.distance` for positive vectors. They take the absolute value of x_i and y_i in the denominator.
- Equivalent to the `canberra` calculator in Mothur, multiplied by the total number of species in `x` and `y`.
- Equivalent to D_{10} in Legendre & Legendre.

Clark's coefficient of divergence involves summing squares and taking a square root afterwards:

$$d(x, y) = \sqrt{\frac{1}{n} \sum_i \left(\frac{x_i - y_i}{x_i + y_i} \right)^2},$$

where n is the number of elements where `x > 0`, `y > 0`, or both. Relation of `clark_coefficient_of_divergence()` to other definitions:

- Equivalent to D_{11} in Legendre & Legendre.

Value

The Canberra distance or Clark's coefficient of divergence. If every element in x and y is zero, Clark's coefficient of divergence is undefined, and we return NaN.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
y <- c(10, 2, 0, 1, 1, 0)
canberra(x, y)
clark_coefficient_of_divergence(x, y)
```

---

chebyshev *Chebyshev distance*

Description

The Chebyshev distance is the maximum absolute difference between the vector elements.

Usage

```
chebyshev(x, y)
```

Arguments

x, y	Numeric vectors

Details

For vectors x and y, the Chebyshev distance is defined as

$$d(x, y) = \max_i |x_i - y_i|.$$

Relation to other definitions:

- Equivalent to the chebyshev() function in scipy.spatial.distance.

Value

The Chebyshev distance between x and y.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
y <- c(10, 2, 0, 1, 1, 0)
chebyshev(x, y) # should be 5
```

correlation_distance *Correlation and cosine distance*

Description

The correlation and cosine distances, which are derived from the dot product of the two vectors.

Usage

```
correlation_distance(x, y)

cosine_distance(x, y)
```

Arguments

x, y	Numeric vectors

Details

For vectors x and y, the cosine distance is defined as the cosine of the angle between the vectors,

$$d(x, y) = 1 - \frac{x \cdot y}{|x||y|},$$

where $|x|$ is the magnitude or L2 norm of the vector, $|x| = \sqrt{\sum_i x_i^2}$. Relation to other definitions:

- Equivalent to the `cosine()` function in `scipy.spatial.distance`.

The correlation distance is simply equal to one minus the Pearson correlation between vectors. Mathematically, it is equivalent to the cosine distance between the vectors after they are centered ($x - \bar{x}$). Relation to other definitions:

- Equivalent to the `correlation()` function in `scipy.spatial.distance`.
- Equivalent to the `1 - mempearson` calculator in Mothur.

Value

The correlation or cosine distance. These are undefined if either x or y contain all zero elements, that is, if $|x| = 0$ or $|y| = 0$. In this case, we return NaN.

Examples

```
x <- c(2, 0)
y <- c(5, 5)
cosine_distance(x, y)
# The two vectors form a 45 degree angle, or pi / 4
1 - cos(pi / 4)

v <- c(3.5, 0.1, 1.4)
w <- c(3.3, 0.5, 0.9)
```

```
correlation_distance(v, w)
1 - cor(v, w)
```

diversity_measures *Diversity measures implemented*

Description

The diversity functions offered in abdiv are organized based on the function signature.

Usage

```
alpha_diversities

beta_diversities

phylogenetic_alpha_diversities

phylogenetic_beta_diversities
```

Format

Four objects of class character.

An object of class character of length 16.

An object of class character of length 42.

An object of class character of length 1.

An object of class character of length 10.

Details

The following character vectors are provided:

- alpha_diversities All non-phylogenetic alpha diversity measures. These functions take a single numeric vector as an argument.
- beta_diversities All non-phylogenetic beta diversity measures. These functions take two numeric vectors as arguments.
- phylogenetic_alpha_diversities There is only one phylogenetic alpha diversity measure implemented, but we use the plural to be consistent with the other vectors. This function takes a numeric vector, a phylogenetic tree object, and optionally a character vector of species labels.
- phylogenetic_beta_diversities Phylogenetic measures of beta diversity. These functions take two numeric vectors, a phylogenetic tree object, and optionally a character vector of species labels.

euclidean *Euclidean and related distances*

Description

These distance and diversity measures are mathematically similar to the Euclidean distance between two vectors.

Usage

```
euclidean(x, y)

rms_distance(x, y)

chord(x, y)

hellinger(x, y)

geodesic_metric(x, y)
```

Arguments

x, y	Numeric vectors

Details

For vectors x and y, the Euclidean distance is defined as

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}.$$

Relation of euclidean() to other definitions:

- Equivalent to R's built-in dist() function with method = "euclidean".
- Equivalent to vegdist() with method = "euclidean".
- Equivalent to the euclidean() function in scipy.spatial.distance.
- Equivalent to the structeuclidean calculator in Mothur, to speciesprofile if x and y are transformed to relative abundance, and to memeuclidean if x and y are transformed to presence/absence.
- Equivalent to D_1 in Legendre & Legendre.
- Equivalent to the *distance between species profiles*, D_{18} in Legendre & Legendre if x and y are transformed to relative abundance.

The *root-mean-square* distance or *average* distance is similar to Euclidean distance. As the name implies, it is computed as the square root of the mean of the squared differences between elements of `x` and `y`:

$$d(x, y) = \sqrt{\frac{1}{n} \sum_i^n (x_i - y_i)^2}.$$

Relation of `rms_distance()` to other definitions:

- Equivalent to D_2 in Legendre & Legendre.

The *chord* distance is the Euclidean distance after scaling each vector by its root sum of squares, $\sqrt{\sum_i x_i^2}$. The chord distance between any two vectors ranges from 0 to $\sqrt{2}$. Relation of `chord()` to other definitions:

- Equivalent to D_3 in Legendre & Legendre.

The *Hellinger* distance is equal to the chord distance computed after a square-root transformation. Relation of `hellinger()` to other definitions:

- Equivalent to D_{17} in Legendre & Legendre.
- Equivalent to the `hellinger` calculator in Mothur.

The *geodesic metric* is a transformed version of the chord distance.

$$d(x, y) = \arccos \left(1 - \frac{d_c^2(x, y)}{2} \right),$$

where d_c is the chord distance. It gives the length of the arc on a hypersphere between the vectors, if the vectors are normalized to unit length. Relation of `geodesic_metric()` to other definitions:

- Equivalent to D_4 in Legendre & Legendre.

Value

The distance between `x` and `y`. The chord distance, Hellinger distance, and geodesic metric are not defined if all elements of either vector are zero. We return `NaN` in this case.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
y <- c(10, 2, 0, 1, 1, 0)
euclidean(x, y)
# The "distance between species profiles"
euclidean(x / sum(x), y / sum(y))
rms_distance(x, y)
chord(x, y)
hellinger(x, y)
# Hellinger is chord distance after square root transform
chord(sqrt(x), sqrt(y))
geodesic_metric(x, y)

# No species in common with x
```

```
v <- c(0, 0, 0, 5, 0, 5)
chord(v, x)
sqrt(2)
```

faith_pd *Faith's phylogenetic diversity*

Description

Faith's phylogenetic diversity gives the total branch length on a phylogenetic tree that is spanned by a community. The abundance of each species in the community is not considered.

Usage

```
faith_pd(x, tree, x_labels = NULL)
```

Arguments

x	A numeric vector of species counts or proportions, or a logical vector of species presence/absence.
tree	A phylogenetic tree object..
x_labels	A character vector of species labels for x.

Details

If the vector x is named, the names will be automatically used to match x with the tree. Missing names are filled in with zero counts. If x is not named and x_labels is provided, these labels are used to match the elements of x with the tree. If x is not named and x_labels is not provided, it is assumed that x is already in the correct order, and we simply check that its length matches the number of tips in the tree.

Value

Faith's phylogenetic diversity, $PD \geq 0$.

References

Faith DP. Conservation evaluation and phylogenetic diversity. Biol. Conserv. 1992;61:1–10. doi: 10.1016/0006-3207(92)91201-3.

Examples

```
# Faith's phylogenetic diversity for whole tree is equal to the sum of the
# branch lengths.
sum(faith_tree$edge.length)
faith_pd(c(1, 1, 1, 1, 1), faith_tree)

# Can use named vector or additional argument to match species to tree.
```

```
faith_tree$tip.label
faith_pd(c(0, 0, 0, 10, 12), faith_tree)
faith_pd(c(d=10, e=12), faith_tree)
faith_pd(c(10, 12), faith_tree, c("d", "e"))
```

faith_tree *Example data for Faith's phylogenetic diversity*

Description

This example was used to illustrate phylogenetic diversity in Faith and Richards (2012).

Usage

```
faith_tree
```

Format

`faith_tree` is a phylogenetic tree with five tips, labeled a-e. It was created with the `ape` library.

Details

In the paper, they give the total branch length as 41, but they don't assign a length to the branch leading to species "b" and "c". Looking at the figure, we estimated that the length should be 4. For that reason, the total branch length of `faith_tree` is 45, rather than 41.

Source

Faith DP, Richards ZT. Biology (Basel). 2012;1(3):906-32. 10.3390/biology1030906

hamming *Hamming distance*

Description

The Hamming distance is the number of positions where the values are different.

Usage

```
hamming(x, y)
```

Arguments

`x, y`	Numeric vectors.

Details

For vectors x and y, the Hamming distance is defined as

$$d(x, y) = \sum_i [x_i \neq y_i],$$

where the quantity in the brackets is 1 if the elements are not equal, and zero if the elements are equal. Relation to other definitions:

- The `hamming()` function in `scipy.spatial.distance` divides the result by the vector length. Our function is equivalent to the SciPy version multiplied by the vector length.
- Equivalent to the `hamming` calculator in Mothur for presence/absence vectors.

Value

The Hamming distance between x and y.

jaccard *Beta diversity for presence/absence data*

Description

These functions transform the input vectors to binary or presence/absence format, then compute a distance or dissimilarity.

Usage

```
jaccard(x, y)

sorenson(x, y)

kulczynski_first(x, y)

kulczynski_second(x, y)

rogers_tanimoto(x, y)

russel_rao(x, y)

sokal_michener(x, y)

sokal_sneath(x, y)

yule_dissimilarity(x, y)
```

Arguments

x, y	Numeric vectors

Details

Many of these indices are covered in Koleff et al. (2003), so we adopt their notation. For two vectors x and y, we define three quantities:

- a is the number of species that are present in both x and y,
- b is the number of species that are present in y but not x,
- c is the number of species that are present in x but not y, and
- d is the number of species absent in both vectors.

The quantity d is seldom used in ecology, for good reason. For details, please see the discussion on the "double zero problem," in section 2 of chapter 7.2 in Legendre & Legendre.

The *Jaccard* index of dissimilarity is $1 - a/(a + b + c)$, or one minus the proportion of shared species, counting over both samples together. Relation of jaccard() to other definitions:

- Equivalent to R's built-in dist() function with method = "binary".
- Equivalent to vegdist() with method = "jaccard" and binary = TRUE.
- Equivalent to the jaccard() function in scipy.spatial.distance, except that we always convert vectors to presence/absence.
- Equivalent to $1 - S_7$ in Legendre & Legendre.
- Equivalent to $1 - \beta_j$, as well as β_{cc}, and β_g in Koleff (2003).

The *Sørenson* or *Dice* index of dissimilarity is $1 - 2a/(2a + b + c)$, or one minus the average proportion of shared species, counting over each sample individually. Relation of sorenson() to other definitions:

- Equivalent to the dice() function in scipy.spatial.distance, except that we always convert vectors to presence/absence.
- Equivalent to the sorclass calculator in Mothur, and to 1 - whittaker.
- Equivalent to $D_{13} = 1 - S_8$ in Legendre & Legendre.
- Equivalent to $1 - \beta_{sor}$ in Koleff (2003). Also equivalent to Whittaker's beta diversity (the second definition, $\beta_w = (S/\bar{a}) - 1$), as well as β_{-1}, β_t, β_{me}, and β_{hk}.

I have not been able to track down the original reference for the first and second Kulczynski indices, but we have good formulas from Legendre & Legendre. The *first Kulczynski index* is $1 - a/(b + c)$, or one minus the ratio of shared to unshared species.

Relation of kulczynski_first to other definitions:

- Equivalent to $1 - S_{12}$ in Legendre & Legendre.
- Equivalent to the kulczynski calculator in Mothur.

Some people refer to the *second Kulczynski index* as the Kulczynski-Cody index. It is defined as one minus the average proportion of shared species in each vector,

$$d = 1 - \frac{1}{2}\left(\frac{a}{a+b} + \frac{a}{a+c}\right).$$

Relation of kulczynski_second to other definitions:

- Equivalent to $1 - S_{13}$ in Legendre & Legendre.
- Equivalent to the `kulczynskicody` calculator in Mothur.
- Equivalent to one minus the Kulczynski similarity in Hayek (1994).
- Equivalent to `vegdist()` with `method = "kulczynski"` and `binary = TRUE`.

The *Rogers-Tanimoto* distance is defined as $(2b+2c)/(a+2b+2c+d)$. Relation of `rogers_tanimoto()` to other definitions:

- Equivalent to the `rogerstanimoto()` function in `scipy.spatial.distance`, except that we always convert vectors to presence/absence.
- Equivalent to $1 - S_2$ in Legendre & Legendre.

The *Russel-Rao* distance is defined $(b + c + d)/(a + b + c + d)$, or the fraction of elements not present in both vectors, counting double absences. Relation of `russel_rao()` to other definitions:

- Equivalent to the `russelrao()` function in `scipy.spatial.distance`, except that we always convert vectors to presence/absence.
- Equivalent to $1 - S_{11}$ in Legendre & Legendre.

The *Sokal-Michener* distance is defined as $(2b+2c)/(a+2b+2c+d)$. Relation of `sokal_michener()` to other definitions:

- Equivalent to the `sokalmichener()` function in `scipy.spatial.distance`, except that we always convert vectors to presence/absence.

The *Sokal-Sneath* distance is defined as $(2b + 2c)/(a + 2b + 2c)$. Relation of `sokal_sneath()` to other definitions:

- Equivalent to the `sokalsneath()` function in `scipy.spatial.distance`, except that we always convert vectors to presence/absence.
- Equivalent to the `anderberg` calculator in Mothur.
- Equivalent to $1 - S_{10}$ in Legendre & Legendre.

The *Yule* dissimilarity is defined as $2bc/(ad + bc)$. Relation of `yule_dissimilarity()` to other definitions:

- Equivalent to the `yule()` function in `scipy.spatial.distance`, except that we always convert vectors to presence/absence.
- Equivalent to $1 - S$, where S is the Yule coefficient in Legendre & Legendre.

Value

The dissimilarity between `x` and `y`, based on presence/absence. The Jaccard, Sorenson, Sokal-Sneath, Yule, and both Kulczynski dissimilarities are not defined if both `x` and `y` have no nonzero elements. In addition, the second Kulczynski index and the Yule index of dissimilarity are not defined if one of the vectors has no nonzero elements. We return `NaN` for undefined values.

`jaccard_components` *Nestedness and turnover components for presence/absence data*

Description

Nestedness and turnover components for presence/absence data

Usage

```
jaccard_turnover(x, y)

jaccard_nestedness(x, y)

sorenson_turnover(x, y)

sorenson_nestedness(x, y)
```

Arguments

`x, y`	Numeric vectors

Value

The nestedness or turnover component of distance between `x` and `y`. This quantity is undefined when either `x` or `y` have no observations, in which case we return `NaN`.

References

Baselga A. Partitioning the turnover and nestedness components of beta diversity. Global Ecol. Biogeogr. 2010;19:134-143.

Baselga A. The relationship between species replacement, dissimilarity derived from nestedness, and nestedness. Global Ecol. Biogeogr. 2012;21:1223–1232.

`kempton_taylor_q` *Kempton-Taylor Q index*

Description

The Kempton-Taylor Q index is designed to measure species in the middle of the abundance distribution.

Usage

```
kempton_taylor_q(x, lower_quantile = 0.25, upper_quantile = 0.75)
```

Arguments

`x`	A numeric vector of species counts or proportions.
`lower_quantile`, `upper_quantile`	Lower and upper quantiles of the abundance distribution. Default values are the ones suggested by Kempton and Taylor.

Details

For a vector of species counts `x`, the Kempton-Taylor Q statistic is equal to the slope of the cumulative abundance curve across a specified quantile range. The cumulative abundance curve is the plot of the number of species against the log-abundance.

Kempton and Taylor originally defined the index as

$$Q = \frac{\frac{1}{2}S}{\log R_2 - \log R_1},$$

where S is the total number of species observed, R_1 is the abundance at the lower quantile, and R_2 is the abundance at the upper quantile. However, this definition only holds if one uses the interquartile range. Because we allow the user to adjust the upper and lower quantiles, we have to find the number of species at these abundance values. Here, we follow the implementation in `scikit-bio` and round inwards to find the quantile values, taking the number of species and log-abundance values at these data points exactly.

- Equivalent to `kempton_taylor_q()` in `skbio.diversity.alpha`.
- Similar to the `qstat` calculator in Mothur. Our implementation differs slightly, and this difference affects the result.

Value

The Kempton-Taylor Q index, $Q < 0$. If the vector sums to zero, we cannot compute the quantiles, and this index is undefined. In that case, we return `NaN`.

References

Kempton RA, Taylor LR. Models and statistics for species diversity. Nature. 1976;262:818-820.

`kullback_leibler_divergence` *Kullback-Leibler divergence*

Description

Kullback-Leibler divergence

Usage

```
kullback_leibler_divergence(x, y)
```

Arguments

x, y	Numeric vectors representing probabilities

Details

Kullback-Leibler divergence is a non-symmetric measure of difference between two probability vectors. In general, KL(x, y) is not equal to KL(y, x).

Because this measure is defined for probabilities, the vectors x and y are normalized in the function so they sum to 1.

Value

The Kullback-Leibler divergence between x and y. We adopt the following conventions if elements of x or y are zero: $0 \log(0/y_i) = 0$, $0 \log(0/0) = 0$, and $x_i \log(x_i/0) = \infty$. As a result, if elements of x are zero, they do not contribute to the sum. If elements of y are zero where x is nonzero, the result will be Inf. If either x or y sum to zero, we are not able to compute the proportions, and we return NaN.

leprieur_tree *Example data for phylogenetic nestedness and turnover components*

Description

This tree was used in Figure 2 of Leprieur et al. (2005) to demonstrate the nestedness and turnover components of phylogenetic beta diversity.

Usage

```
leprieur_tree
```

Format

leprieur_tree is a phylogenetic tree with 8 tips, labeled a-h. It was created with the ape library. All edges (branches) in the tree are of length 1.

Source

Leprieur F, Albouy C, De Bortoli J, Cowman PF, Bellwood DR, Mouillot D. Quantifying phylogenetic beta diversity: distinguishing between "true" turnover of lineages and phylogenetic diversity gradients. PLoS One. 2012;7(8):e42760. 10.1371/journal.pone.0042760

lozupone_tree *Example data for UniFrac distance*

Description

This example was used to illustrate unweighted UniFrac distance in Lozupone and Knight (2005).

Usage

```
lozupone_tree

lozupone_panel_a

lozupone_panel_b
```

Format

`lozupone_tree` is a phylogenetic tree with 14 tips, labeled A-N. It was created with the `ape` library.

The data frames `lozupone_panel_a` and `lozupone_panel_b` are transcribed from Figure 1 of the paper. They have the following columns:

Species The species label, matching to the phylogenetic tree.

SampleID The community, either "Circle" or "Square".

Counts The number of organisms counted per species, always 1 for this example.

An object of class `data.frame` with 14 rows and 3 columns.

An object of class `data.frame` with 14 rows and 3 columns.

Source

Lozupone C, Knight R. UniFrac: a new phylogenetic method for comparing microbial communities. Applied and environmental microbiology. 2005;71(12):8228–8235. 10.1128/AEM.71.12.8228-8235.2005

manhattan *Manhattan and related distances*

Description

The Manhattan or city block distance is the sum of absolute differences between the elements of two vectors. The *mean character* difference is a closely related measure.

Usage

```
manhattan(x, y)

mean_character_difference(x, y)

modified_mean_character_difference(x, y)
```

Arguments

x, y	Numeric vectors

Details

For vectors x and y, the Manhattan distance is given by

$$d(x, y) = \sum_i |x_i - y_i|.$$

Relation of manhattan() to other definitions:

- Equivalent to R's built-in dist() function with method = "manhattan".
- Equivalent to vegdist() with method = "manhattan".
- Equivalent to the cityblock() function in scipy.spatial.distance.
- Equivalent to the manhattan calculator in Mothur.
- Equivalent to D_7 in Legendre & Legendre.
- Whittaker's index of association (D_9 in Legendre & Legendre) is the Manhattan distance computed after transforming to proportions and dividing by 2.

The mean character difference is the Manhattan distance divided by the length of the vectors. It was proposed by Cain and Harrison in 1958. Relation of mean_character_difference() to other definitions:

- Equivalent to D_8 in Legendre & Legendre.
- For binary data, equivalent to $1 - S_1$ in Legendre & Legendre, where S_1 is the simple matching coefficient.

The modified mean character difference is the Manhattan distance divided by the number elements where either x or y (or both) are nonzero. Relation of modified_mean_character_difference() to other definitions:

- Equivalent to D_{19} in Legendre & Legendre.
- Equivalent to vegdist() with method = "altGower".
- For binary data, it is equivalent to the Jaccard distance.

Value

The distance between x and y. The modified mean character difference is undefined if all elements in x and y are zero, in which case we return NaN.

References

Cain AJ, Harrison GA. An analysis of the taxonomist's judgment of affinity. Proceedings of the Zoological Society of London 1958;131:85-98.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
y <- c(10, 2, 0, 1, 1, 0)
manhattan(x, y)
# Whittaker's index of association
manhattan(x / sum(x), y / sum(y)) / 2

mean_character_difference(x, y)
# Simple matching coefficient for presence/absence data
# Should be 2 / 6
mean_character_difference(x > 0, y > 0)

modified_mean_character_difference(x, y)
# Jaccard distance for presence/absence data
modified_mean_character_difference(x > 0, y > 0)
jaccard(x, y)
```

margalef *Margalef's richness index*

Description

Margalef's richness index

Usage

```
margalef(x)
```

Arguments

x	A numeric vector of species counts.

Details

For a vector x of species counts, Margalef's index is

$$D = \frac{S - 1}{\log N},$$

where S is the total number of species observed and N is the total number of counts.

This index is appropriate only for raw counts, not transformed counts or proportions.

Equivalent to `margalef()` in `skbio.diversity.alpha`.

Value

The value of Margalef's index, $D \geq 0$. This index is undefined when the total number of counts is 1 or 0, in which case we return NaN.

References

Margalef R. Information theory in ecology. General Systems 3. 1958;36-71.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
margalef(x)
```

match_to_tree *Match vector of counts to phylogenetic tree*

Description

Match vector of counts to phylogenetic tree

Usage

```
match_to_tree(x, tree, x_labels = NULL)
```

Arguments

x	A vector of species counts.
tree	A phylogenetic tree of class "phylo".
x_labels	A vector of species labels for x.

Details

This function applies a couple of different methods to arrange the data in x to match a phylogenetic tree.

- If x_labels is provided, we use this vector to match the elements of x with the tip labels in the tree.
- If x_labels is not provided and x is a named vector, we use the names to match the tip labels in the tree.
- If x_labels is not provided and x is not named, we assume that x is already in the correct order, check that the length of x matches the number of tips in the tree, and return x.

Value

The vector x, re-arranged to match the tree.

mcintosh_d *McIntosh dominance index D*

Description

McIntosh dominance index D

Usage

```
mcintosh_d(x)
```

Arguments

x	A numeric vector of species counts.

Details

For a vector x of raw species counts, the McIntosh dominance index is defined as

$$D = \frac{N - U}{N - \sqrt{N}},$$

where N is the total number of counts and $U = \sqrt{\sum_i x_i^2}$.

This index is appropriate only for raw counts, not transformed counts or proportions.

Equivalent to `mcintosh_d()` in `skbio.diversity.alpha`.

Value

The McIntosh dominance index, $0 \leq D < 1$. The index is undefined when the total number of counts is 1 or 0, in which case we return `NaN`.

References

McIntosh RP. An index of diversity and the relation of certain concepts to diversity. Ecology. 1967;48:1115-1126.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
mcintosh_d(x)
```

mcintosh_e *McIntosh's evenness measure E*

Description

McIntosh's evenness measure E

Usage

```
mcintosh_e(x)
```

Arguments

x	A numeric vector of species counts.

Details

For a vector x of raw species counts, the McIntosh evenness measure is

$$E = \frac{\sqrt{\sum_i x_i^2}}{\sqrt{(N - S + 1)^2 + S - 1},}$$

where N is the total number of counts and S is the total number of species observed.

This index is appropriate only for raw counts, not transformed counts or proportions.

Equivalent to `mcintosh_e()` in `skbio.diversity.alpha`.

Value

McIntosh's evenness measure, $0 < E \leq 1$. The index is undefined when the total number of counts is 0, in which case we return NaN.

References

Heip C, Engels P. Comparing Species Diversity and Evenness Indices. J. Mar. Bioi. Ass. U.K. 1974;54:559-563.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
mcintosh_e(x)
```

`menhinick` *Menhinick's richness index*

Description

Menhinick's richness index

Usage

```
menhinick(x)
```

Arguments

`x`	A numeric vector of species counts.

Details

For a vector `x` of raw species counts, the Menhinick's richness index is $\frac{S}{\sqrt{N}}$, where N is the total number of counts and S is the total number of species observed.

This index is appropriate only for raw counts, not transformed counts or proportions.

Equivalent to `menhinick()` in `skbio.diversity.alpha`.

Value

Menhinick's richness index, $R > 0$. The index is undefined when the total number of counts is 0, in which case we return `NaN`.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
menhinick(x)
```

`minkowski` *Minkowski distance*

Description

The Minkowski metric is a generalized form of Euclidean (p=2) and Manhattan (p=1) distance.

Usage

```
minkowski(x, y, p = 1)
```

Arguments

`x, y`	Numeric vectors.
`p`	Exponent parameter, a single number greater than zero.

Details

For vectors x and y, the Minkowski distance is defined as

$$d(x, y) = \left(\sum_i |x_i - y_i|^p \right)^{1/p}.$$

Relation to other definitions:

- Equivalent to R's built-in `dist()` function with `method = "minkowski"`.
- Equivalent to the `minkowski()` function in `scipy.spatial.distance`.
- Equivalent to D_6 in Legendre & Legendre.

The default value of `p = 1` makes this distance equal to the Manhattan distance.

Value

The Minkowski distance between x and y.

morisita *The Morisita index and Horn-Morisita index*

Description

The Morisita and the Horn-Morisita indices measure the probability that individuals drawn one from each vector will belong to different species, relative to drawing from each vector separately. The Morisita index is formulated for count data only, whereas the Horn-Morisita index can be used with transformed counts or proportions.

Usage

```
morisita(x, y)

horn_morisita(x, y)
```

Arguments

x, y	Numeric vectors

Details

For two vectors x and y, the Morisita index of dissimilarity is

$$d(x, y) = 1 - \frac{2 \sum_i x_i y_i}{(\lambda_x + \lambda_y) N_x N_y},$$

where

$$\lambda_x = \frac{\sum_i x_i (x_i - 1)}{N_x (N_x - 1)}$$

and $N_x = \sum_i x_i$ The formula for λ_x is the unbiased estimate for the probability of drawing two individuals of the same species from x, without replacement. The correction for sampling without replacement only makes sense for species count data.

Relation of `morisita()` to other definitions:

- Equivalent to `vegdist()` with `method = "morisita"`.

Horn (1966) reformulated the index to use the equations for sampling with replacement in λ_x and λ_y:

$$\lambda_x = \frac{\sum_i x_i^2}{N_x^2}$$

With this modification, the index is valid for proportions or transformed count data.

Relation of `horn_morisita()` to other definitions:

- Equivalent to `vegdist()` with `method = "horn"`.
- Equivalent to the `morisitahorn` calculator in Mothur.

Value

The Morisita or Horn-Morisita index between `x` and `y`. Both are undefined if `x` or `y` have no nonzero elements, in which case we return `NaN`.

References

Mosrisita M. Measuring of interspecific association and similarity between communities. Memoirs of the Faculty of Science, Kyushu Univ., Series E (Biology). 1959;3:65-80.

Horn HS. Measurement of "Overlap" in Comparative Ecological Studies. The American Naturalist, 1966;100(914):419-424.

`richness` *Richness or number of observed species*

Description

Richness or number of observed species

Usage

```
richness(x)
```

Arguments

`x`	A numeric vector of species counts or proportions.

Details

The richness is simply the number of nonzero elements in x. Relation to other definitions:

- Equivalent to observed_otus() in skbio.diversity.alpha.
- Equivalent to specnumber in vegan.
- Equivalent to the sobs calculator in Mothur.

Value

The number of species observed, $R \geq 0$.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
richness(x) # 4
```

---

ruzicka *Ruzicka or weighted Jaccard distance*

Description

Ruzicka or weighted Jaccard distance

Usage

```
ruzicka(x, y)
```

Arguments

x, y	Numeric vectors.

Details

For vectors x and y, the Ruzicka distance is defined as

$$d(x, y) = 1 - \frac{\sum_i \min(x, y)}{\sum_i \max(x, y)}.$$

Relation to other definitions:

- Equivalent to vegdist() with method = "jaccard".
- Related to the Bray-Curtis distance, $d_r = 2d_{bc}/(1 + d_{bc})$.

Value

The Ruzicka distance between x and y. The distance is not defined if all elements in x and y are zero, and we return NaN in this case.

shannon *Shannon diversity and related measures*

Description

The Shannon index of diversity

Usage

```
shannon(x, base = exp(1))

brillouin_d(x)

heip_e(x)

pielou_e(x)
```

Arguments

x	A numeric vector of species counts or proportions.
base	Base of the logarithm to use in the calculation.

Details

The Shannon index of diversity or Shannon information entropy has deep roots in information theory. It is defined as

$$H = -\sum_i p_i \log p_i,$$

where p_i is the species proportion. Relation to other definitions:

- Equivalent to `diversity()` in `vegan` with `index = "shannon"`.
- Equivalent to `shannon()` in `skbio.diversity.alpha`.

The Brillouin index (Brillouin 1956) is similar to Shannon's index, but accounts for sampling without replacement. For a vector of species counts `x`, the Brillouin index is

$$\frac{1}{N} \log \frac{N!}{\prod_i x_i!} = \frac{\log N! - \sum_i \log x_i!}{N}$$

where N is the total number of counts. Relation to other definitions:

- Equivalent to `brillouin_d()` in `skbio.diversity.alpha`.
- Equivalent to the `shannon` calculator in Mothur.

The Brillouin index accounts for the total number of individuals sampled, and should be used on raw count data, not proportions.

Heip's evenness measure is

$$\frac{e^H - 1}{S - 1},$$

where S is the total number of species observed. Relation to other definitions:

- Equivalent to `heip_e()` in `skbio.diversity.alpha`.

Pielou's Evenness index $J = H / \log S$. Relation to other definitions:

- Equivalent to `peilou_e()` in `skbio.diversity.alpha`.

Value

The Shannon diversity, $H \geq 0$, or related quantity. The value of H is undefined if `x` sums to zero, and we return `NaN` in this case. Heip's evenness measure and Pielou's Evenness index are undefined if only one element of `x` is nonzero, and again we return `NaN` if this is the case.

References

Brillouin L. Science and Information Theory. 1956;Academic Press, New York.

Pielou EC. The Measurement of Diversity in Different Types of Biological Collections. Journal of Theoretical Biology. 1966;13:131-144.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
shannon(x)

# Using a different base is the same as dividing by the log of that base
shannon(x, base = 10)
shannon(x) / log(10)

brillouin_d(x)

# Brillouin index should be almost identical to Shannon index for large N
brillouin_d(10000 * x)
shannon(10000 * x)

heip_e(x)
(exp(shannon(x)) - 1) / (richness(x) - 1)

pielou_e(x)
shannon(x) / log(richness(x))
```

simpson *Simpson's index and related measures*

Description

These measures are based on the sum of squared species proportions. The function dominance() gives this quantity, simpson() gives one minus this quantity, invsimpson() gives the reciprocal of the quantity, and simpson_e gives the reciprocal divided by the number of species.

Usage

```
simpson(x)

dominance(x)

invsimpson(x)

simpson_e(x)
```

Arguments

x	A numeric vector of species counts or proportions.

Details

For a vector of species counts x, the dominance index is defined as

$$D = \sum_i p_i^2,$$

where p_i is the species proportion, $p_i = x_i/N$, and N is the total number of counts. This is equal to the probability of selecting two individuals from the same species, with replacement. Relation to other definitions:

- Equivalent to dominance() in skbio.diversity.alpha.
- Similar to the simpson calculator in Mothur. They use the unbiased estimate $p_i = x_i(x_i - 1)/(N(N-1))$.

Simpson's index is defined here as $1 - D$, or the probability of selecting two individuals from different species, with replacement. Relation to other definitions:

- Equivalent to diversity() in vegan with index = "simpson".
- Equivalent to simpson() in skbio.diversity.alpha.

The inverse Simpson index is $1/D$. Relation to other definitions:

- Equivalent to diversity() in vegan with index = "invsimpson".
- Equivalent to enspie() in skbio.diversity.alpha.

- Similar to the `invsimpson` calculator in Mothur. They use the unbiased estimate $p_i = x_i(x_i - 1)/(N(N-1))$.

Simpson's evenness index is the inverse Simpson index divided by the number of species observed, $1/(DS)$. Relation to other definitions:

- Equivalent to `simpson_e()` in `skbio.diversity.alpha`.

Please be warned that the naming conventions vary between sources. For example Wikipedia calls D the Simpson index and $1 - D$ the Gini-Simpson index. We have followed the convention from `vegan`, to avoid confusion within the `R` ecosystem.

Value

The value of the dominance ($0 < D \leq 1$), Simpson index, or inverse Simpson index. The dominance is undefined if the vector sums to zero, in which case we return `NaN`.

Examples

```
x <- c(15, 6, 4, 0, 3, 0)
dominance(x)

# Simpson is 1 - D
simpson(x)
1 - dominance(x)

# Inverse Simpson is 1/D
invsimpson(x)
1 / dominance(x)

# Simpson's evenness is 1 / (D * S)
simpson_e(x)
1 / (dominance(x) * richness(x))
```

`strong` *Strong's dominance index*

Description

Strong's dominance index measures the maximum departure between the observed proportions and a perfectly even community.

Usage

```
strong(x)
```

Arguments

`x`	A numeric vector of species counts.

Details

Strong's dominance index is defined as

$$D_W = \max_i \left[\frac{b_i}{N} - \frac{i}{S} \right],$$

where b_i is the abundance of the ith species, ordered from smallest to largest, N is the total number of counts, and S is the number of species observed.

Equivalent to `strong()` in `skbio.diversity.alpha`.

Value

Strong's dominance index, $0 \leq D_W < 1$. The index is undefined if `x` sums to 0, and we return `NaN` in this case.

References

Strong WL. Assessing species abundance unevenness within and between plant communities. Community Ecology. 2002;3:237-246.

Examples

```
x <- c(9, 0, 1, 2, 5, 2, 1, 1, 0, 7, 2, 1, 0, 1, 1)
strong(x)
```

unifrac *UniFrac distance*

Description

The UniFrac distance is a phylogenetically-weighted distance between two communities of organisms. The measure has been extended a number of times to include abundance-weighted and variance-adjusted versions.

Usage

```
unweighted_unifrac(x, y, tree, xy_labels = NULL)

weighted_unifrac(x, y, tree, xy_labels = NULL)

weighted_normalized_unifrac(x, y, tree, xy_labels = NULL)

variance_adjusted_unifrac(x, y, tree, xy_labels = NULL)

generalized_unifrac(x, y, tree, alpha = 0.5, xy_labels = NULL)

information_unifrac(x, y, tree, xy_labels = NULL)

phylosor(x, y, tree, xy_labels = NULL)
```

Arguments

`x, y`	Numeric vectors of species counts or proportions.
`tree`	A phylogenetic tree object.
`xy_labels`	A character vector of species labels for `x` and `y`.
`alpha`	Generalized UniFrac parameter.

Details

These functions compute different variations of the UniFrac distance between communities described by the vectors `x` and `y`. If the vectors are named, the names will be automatically used to match the vectors with the tree. Missing names are filled in with zero counts. If the vectors are not named and `xy_labels` is provided, these labels will be used to match the vectors with the tree. If the vectors are not named and `xy_labels` is not provided, it is assumed that the vectors are already in the correct order, and we simply check that their length matches the number of tips in the tree.

`unweighted_unifrac` gives the original UniFrac distance from Lozupone and Knight (2005), which is the fraction of total branch length leading to community `x` or community `y`, but not both. It is based on species presence/absence.

`weighted_unifrac` gives the abundance-weighted version of UniFrac proposed by Lozupone et al. (2007). In this measure, the branch lengths of the tree are multiplied by the absolute difference in species abundances below each branch.

`weighted_normalized_unifrac` provides a normalized version of `weighted_unifrac`, so the distance is between 0 and 1.

`variance_adjusted_unifrac` was proposed by Chang et al. (2011) to adjust for the variation of weights in weighted UniFrac under random sampling.

`generalized_unifrac` was proposed by Chen et al. (2012) to provide a unified mathematical framework for weighted and unweighted UniFrac distance. It includes a parameter, α, which can be used to adjust the abundance-weighting in the distance. A value of $\alpha = 1$ corresponds to weighted UniFrac. A value of $\alpha = 0$ corresponds to unweighted UniFrac if presence/absence vectors are provided. The authors suggest a value of $\alpha = 0.5$ as a compromise between weighted and unweighted distances.

`information_unifrac` was proposed by Wong et al. (2016) to connect UniFrac distance with compositional data analysis. They also proposed a "ratio UniFrac" distance, which is not yet implemented.

`phylosor`, proposed by Bryant et al. (2008), is closely related to unweighted UniFrac distance. If unweighted UniFrac distance is the analogue of Jaccard distance using branches on a phylogenetic tree, PhyloSor is the analogue of Sorenson dissimilarity.

Value

The UniFrac distance between communities `x` and `y`. The distance is not defined if either `x` or `y` have all zero elements. We return `NaN` if this is the case.

References

Lozupone C, Knight R. UniFrac: a new phylogenetic method for comparing microbial communities. Applied and environmental microbiology. 2005;71(12):8228–8235. 10.1128/AEM.71.12.8228-8235.2005

Lozupone CA, Hamady M, Kelley ST, Knight R. Quantitative and qualitative β diversity measures lead to different insights into factors that structure microbial communities. Applied and environmental microbiology. 2007;73(5):1576–1585. 10.1128/AEM.01996-06

Chang Q., et al. Variance adjusted weighted UniFrac: a powerful beta diversity measure for comparing communities based on phylogeny. BMC Bioinformatics. 2011;12:118. 10.1186/1471-2105-12-118

Chen J, Bittinger K, Charlson ES, Hoffmann C, Lewis J, Wu GD, et al. Associating microbiome composition with environmental covariates using generalized UniFrac distances. Bioinformatics. 2012;28(16):2106–2113. 10.1093/bioinformatics/bts342

Wong RG, Wu JR, Gloor GB. Expanding the UniFrac Toolbox. PLOS ONE. 2016;11(9):1–20. 10.1371/journal.pone.0161196

Bryant JA, Lamanna C, Morlon H, Kerkhoff AJ, Enquist BJ, Green JL. Microbes on mountainsides: contrasting elevational patterns of bacterial and plant diversity. Proc Natl Acad Sci U S A. 2008;105 Suppl 1:11505-11. 10.1073/pnas.0801920105

Examples

```
# From Lozupone and Knight (2005), Figure 1.
# Panel A
x1 <- c(1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1)
x2 <- c(0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0)
unweighted_unifrac(x1, x2, lozupone_tree)

# Panel B
x3 <- c(0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1)
x4 <- c(1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0)
unweighted_unifrac(x3, x4, lozupone_tree)

# Can use named vectors to specify species
weighted_normalized_unifrac(
  c(A=1, C=1, D=1, F=1, I=1, L=1, N=1),
  c(B=1, E=1, G=1, H=1, J=1, K=1, M=1),
  lozupone_tree)
weighted_normalized_unifrac(x1, x2, lozupone_tree)

# Generalized UniFrac is equal to weighted normalized UniFrac when alpha = 1
generalized_unifrac(x1, x2, lozupone_tree, alpha=1)
generalized_unifrac(x1, x2, lozupone_tree, alpha=0.5)
```

`unifrac_components` *Nestedness and turnover components of unweighted UniFrac distance*

Description

Nestedness and turnover components of unweighted UniFrac distance

Usage

```
unweighted_unifrac_turnover(x, y, tree, xy_labels = NULL)

unweighted_unifrac_nestedness(x, y, tree, xy_labels = NULL)

phylosor_turnover(x, y, tree, xy_labels = NULL)

phylosor_nestedness(x, y, tree, xy_labels = NULL)
```

Arguments

`x, y`	Numeric vectors of species counts or proportions.
`tree`	A phylogenetic tree object.
`xy_labels`	A character vector of species labels for `x` and `y`.

Details

Leprieur et al. (2012) showed that measures of phylogenetic beta diversity could be partitioned into nestedness and turnover components, following the approach of Baselga (2010) for Sorenson dissimilarity.

Value

The nestedness or turnover component of the UniFrac distance between communities `x` and `y`. This quantity is undefined when either `x` or `y` have all elements equal to zero, in which case we return `NaN`.

References

Baselga A. Partitioning the turnover and nestedness components of beta diversity. Global Ecol. Biogeogr. 2010;19:134-143.

Leprieur F, Albouy C, De Bortoli J, Cowman PF, Bellwood DR, Mouillot D. Quantifying phylogenetic beta diversity: distinguishing between "true" turnover of lineages and phylogenetic diversity gradients. PLoS One. 2012;7(8):e42760. 10.1371/journal.pone.0042760

Examples

```
# Vectors x and y have turnover but no nestedness
x <- c(1, 1, 1, 0, 0, 0, 0, 0)
y <- c(0, 1, 1, 1, 0, 0, 0, 0)

unweighted_unifrac(x, y, leprieur_tree)
unweighted_unifrac_turnover(x, y, leprieur_tree)
unweighted_unifrac_nestedness(x, y, leprieur_tree)
```

```
phylosor(x, y, leprieur_tree)
phylosor_turnover(x, y, leprieur_tree)
phylosor_nestedness(x, y, leprieur_tree)

# Vectors y and z have nestedness but no turnover
z <- c(0, 1, 1, 1, 1, 1, 1, 1)

unweighted_unifrac(y, z, leprieur_tree)
unweighted_unifrac_turnover(y, z, leprieur_tree)
unweighted_unifrac_nestedness(y, z, leprieur_tree)

phylosor(y, z, leprieur_tree)
phylosor_turnover(y, z, leprieur_tree)
phylosor_nestedness(y, z, leprieur_tree)
```

weighted_kulczynski_second *Weighted Kulczynski distance*

Description

The quantitative version of the second Kulczynski index

Usage

```
weighted_kulczynski_second(x, y)
```

Arguments

x, y	Numeric vectors

Details

The quantitative version of the second Kulczynski index is defined as

$$d(x, y) = 1 - \frac{1}{2} \left(\frac{\sum_i \min{(x_i, y_i)}}{\sum_i x_i} + \frac{\sum_i \min{(x_i, y_i)}}{\sum_i y_i} \right).$$

Relation of `weighted_kulczynski_second()` to other definitions:

- Equivalent to `vegdist()` with `method = "kulczynski"`.
- Equivalent to `structkulczynski` in Mothur.
- Equivalent to $1 - S_{18}$ in Legendre & Legendre.

Value

The weighted Kulczynski distance between `x` and `y`. The distance is undefined if the sum of `x` or the sum of `y` is zero, in which case we return `NaN`.

Index

∗ **datasets**
    diversity_measures, 11
    faith_tree, 15
    leprieur_tree, 21
    lozupone_tree, 22

abundance_jaccard, 2
abundance_sorenson *(*abundance_jaccard*)*, 2
alpha_diversities *(*diversity_measures*)*, 11

berger_parker_d, 4
beta_diversities *(*diversity_measures*)*, 11
binomial_deviance, 4
bray_curtis, 6
bray_curtis_balanced *(*bray_curtis_components*)*, 7
bray_curtis_components, 7
bray_curtis_gradient *(*bray_curtis_components*)*, 7
brillouin_d *(*shannon*)*, 32

canberra, 8
chebyshev, 9
chord *(*euclidean*)*, 12
clark_coefficient_of_divergence *(*canberra*)*, 8
correlation_distance, 10
cosine_distance *(*correlation_distance*)*, 10
cy_dissimilarity *(*binomial_deviance*)*, 4

diversity_measures, 11
dominance *(*simpson*)*, 34

euclidean, 12

faith_pd, 14
faith_tree, 15

generalized_unifrac *(*unifrac*)*, 36
geodesic_metric *(*euclidean*)*, 12

hamming, 15
heip_e *(*shannon*)*, 32
hellinger *(*euclidean*)*, 12
horn_morisita *(*morisita*)*, 29

information_unifrac *(*unifrac*)*, 36
invsimpson *(*simpson*)*, 34

jaccard, 16
jaccard_components, 19
jaccard_nestedness *(*jaccard_components*)*, 19
jaccard_turnover *(*jaccard_components*)*, 19

kempton_taylor_q, 19
kulczynski_first *(*jaccard*)*, 16
kulczynski_second *(*jaccard*)*, 16
kullback_leibler_divergence, 20

leprieur_tree, 21
lozupone_panel_a *(*lozupone_tree*)*, 22
lozupone_panel_b *(*lozupone_tree*)*, 22
lozupone_tree, 22

manhattan, 22
margalef, 24
match_to_tree, 25
mcintosh_d, 26
mcintosh_e, 27
mean_character_difference *(*manhattan*)*, 22
menhinick, 28
minkowski, 28
modified_mean_character_difference *(*manhattan*)*, 22
morisita, 29

phylogenetic_alpha_diversities
(diversity_measures), 11
phylogenetic_beta_diversities
(diversity_measures), 11
phylosor (unifrac), 36
phylosor_nestedness
(unifrac_components), 38
phylosor_turnover (unifrac_components),
38
pielou_e (shannon), 32

richness, 30
rms_distance (euclidean), 12
rogers_tanimoto (jaccard), 16
russel_rao (jaccard), 16
ruzicka, 31
ruzicka_balanced
(bray_curtis_components), 7
ruzicka_gradient
(bray_curtis_components), 7

shannon, 32
simpson, 34
simpson_e (simpson), 34
sokal_michener (jaccard), 16
sokal_sneath (jaccard), 16
sorenson (jaccard), 16
sorenson_nestedness
(jaccard_components), 19
sorenson_turnover (jaccard_components),
19
strong, 35

unifrac, 36
unifrac_components, 38
unweighted_unifrac (unifrac), 36
unweighted_unifrac_nestedness
(unifrac_components), 38
unweighted_unifrac_turnover
(unifrac_components), 38

variance_adjusted_unifrac (unifrac), 36

weighted_kulczynski_second, 40
weighted_normalized_unifrac (unifrac),
36
weighted_unifrac (unifrac), 36

yule_dissimilarity (jaccard), 16