

Package ‘hilldiv3’

October 6, 2026

Title Integral Analysis of Diversity Based on Hill Numbers

Version 3.0.0

Description Measures and compares the diversity of biological communities (e.g. tables of operational taxonomic units (OTUs), amplicon sequence variants (ASVs) or metagenome-assembled genomes (MAGs)) based on Hill numbers, in a unified framework for neutral, phylogenetic and functional diversity measurement, diversity partitioning, (dis)similarity measurement, diversity profiles, evenness and redundancy. The statistical framework encompasses richness, Shannon and Simpson diversity, Faith's phylogenetic diversity (PD), Rao's quadratic entropy and Sorensen- and UniFrac-type dissimilarities, all grounded in a single Hill-number framework. Methods are described in Jost (2007) <doi:10.1890/06-1736.1>, Chao et al. (2010) <doi:10.1098/rstb.2010.0272>, Chiu et al. (2014) <doi:10.1890/12-0960.1> and reviewed in Alberdi & Gilbert (2019) <doi:10.1111/1755-0998.13014>. Optional import adapters interoperate with the Bioconductor packages 'phyloseq', 'SummarizedExperiment' and 'TreeSummarizedExperiment', which are available from <<https://bioconductor.org>>.

License GPL-3

Language en-GB

URL <https://github.com/alberdilab/hilldiv3>,
<https://alberdilab.github.io/hilldiv3/>

BugReports <https://github.com/alberdilab/hilldiv3/issues>

Depends R (>= 4.1.0)

Imports ape, cli, grDevices, graphics, methods, rlang, stats, utils

Suggests cluster, furrr, future, ggplot2, knitr, patchwork, phyloseq, progressr, rmarkdown, SummarizedExperiment, testthat (>= 3.0.0), tibble, TreeSummarizedExperiment, vegan

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true
RoxygenNote 7.3.3
NeedsCompilation no
Author Antton Alberdi [aut, cre] (ORCID:
 <<https://orcid.org/0000-0002-2875-6446>>)
Maintainer Antton Alberdi <antton.alberdi@sund.ku.dk>
Repository CRAN
Date/Publication 2026-10-06 16:40:02 UTC

Contents

gut_counts	2
gut_traits	3
gut_tree	4
hilldiss	4
hilldiv	5
hilleven	7
hillpair	8
hillpart	10
hillprof	12
hillred	13
hillsim	14
match_data	16
plot.hill_profile	16
plot.hill_redundancy	17
traits2dist	17
tss	18
Index	19

gut_counts	<i>Simulated gut microbiome MAG count table</i>
------------	---

Description

A small, **simulated** example data set representing metagenome-assembled genome (MAG) abundances across host samples from two groups (control and treatment), where a block of MAGs is enriched in the treatment group so that beta diversity is non-trivial. Generated by `data-raw/make-data.R`.

Usage

gut_counts

Format

An integer matrix with 24 rows (MAGs, mag01..mag24) and 12 columns (samples, ctrl01..ctrl06 and trt01..trt06).

See Also

[gut_tree](#), [gut_traits](#)

Examples

```
hilldiv(gut_counts, q = c(0, 1, 2))
```

gut_traits

Functional traits for the simulated gut MAGs

Description

A trait table for the 24 MAGs in [gut_counts](#), mixing continuous, categorical and binary traits. Convert it to a functional distance with [traits2dist\(\)](#) for the functional-diversity paths.

Usage

```
gut_traits
```

Format

A data frame with 24 rows (MAGs) and 4 columns:

genome_size Approximate genome size in Mbp (numeric).

gc_content GC content as a proportion (numeric).

oxygen Oxygen tolerance: aerobe, anaerobe or facultative (factor).

motility Motility indicator, 0/1 (integer).

See Also

[gut_counts](#), [gut_tree](#), [traits2dist\(\)](#)

Examples

```
d <- traits2dist(gut_traits)
hilldiv(gut_counts, q = c(0, 1), dist = d)
```

gut_tree

Phylogeny for the simulated gut MAGs

Description

An ultrametric coalescent tree over the 24 MAGs in [gut_counts](#), scaled to unit depth. Use it for the phylogenetic-diversity paths.

Usage

```
gut_tree
```

Format

A phylo object (see the **ape** package) with 24 tips whose labels match the rows of [gut_counts](#).

See Also

[gut_counts](#), [gut_traits](#)

Examples

```
hilldiv(gut_counts, q = c(0, 1), tree = gut_tree)
```

hilldiss

Hill numbers-based dissimilarity

Description

Compute overall (multi-sample) dissimilarity metrics from the Hill-number beta diversity following Chiu et al. (2014). These are the complements of the similarities returned by [hillsim\(\)](#).

Usage

```
hilldiss(
  data,
  q = c(0, 1, 2),
  metric = c("S", "C", "U", "V"),
  tree = NULL,
  dist = NULL,
  tau = NULL,
  type = c("auto", "neutral", "phylogenetic", "functional"),
  out = c("tibble", "matrix")
)
```

Arguments

data	A count table (taxa x samples) or a supported object; a single sample is not meaningful for partitioning.
q	Numeric vector of diversity orders (≥ 0). Defaults to <code>c(0, 1, 2)</code> (richness, Shannon, Simpson).
metric	Dissimilarity metric(s) to return, any of "S", "C", "U", "V". Defaults to all four.
tree	A phylogenetic tree of class phylo whose tip labels match the taxa in data.
dist	A functional distance matrix (or dist) over the taxa.
tau	Optional functional distance threshold. Defaults to <code>max(dist)</code> .
type	Diversity type: "auto" (default) infers it from the inputs (counts only -> neutral, +tree -> phylogenetic, +dist -> functional); an explicit "neutral", "phylogenetic" or "functional" asserts the type and is validated against the inputs (e.g. "phylogenetic" requires a tree; "neutral" ignores any tree/dist carried by the object).
out	Output shape: "tibble" (default) returns a long-format data.frame with columns q, metric, value; "matrix" returns the legacy matrix (orders in rows, metrics in columns, dropped to a vector for a single metric).

Value

A long-format data.frame of class `hill_dissimilarity` (default, with a `plot()` method), or a matrix/vector of dissimilarities when `out = "matrix"`.

See Also

[hillsim\(\)](#), [hillpair\(\)](#), [hillpart\(\)](#)

Examples

```
counts <- matrix(c(10, 0, 5, 2, 8, 1), nrow = 3,
                 dimnames = list(c("t1", "t2", "t3"), c("s1", "s2")))
hilldiss(counts)
plot(hilldiss(counts))
```

hilldiv

Hill numbers computation

Description

Compute neutral, phylogenetic and/or functional Hill numbers (alpha diversity) from a single sample or a count table. By default the computation is *cumulative*: every diversity type whose inputs are present is returned. Counts are always available, so neutral is always computed; a tree adds phylogenetic and a dist adds functional. Supplying both a tree and a dist therefore returns neutral, phylogenetic and functional side by side in a single tibble (with a type column). Use type to restrict the output to a subset.

Usage

```
hilldiv(
  data,
  q = c(0, 1, 2),
  tree = NULL,
  dist = NULL,
  tau = NULL,
  type = c("auto", "neutral", "phylogenetic", "functional"),
  reference = c("pool", "sample"),
  out = c("tibble", "matrix")
)
```

Arguments

data	Counts: a numeric vector (one sample), a matrix/data.frame (taxa x samples), a phyloseq object or a TreeSummarizedExperiment.
q	Numeric vector of diversity orders (≥ 0). Defaults to <code>c(0, 1, 2)</code> (richness, Shannon, Simpson).
tree	A phylogenetic tree of class <code>phylo</code> whose tip labels match the taxa in data.
dist	A functional distance matrix (or <code>dist</code>) over the taxa.
tau	Optional functional distance threshold. Defaults to <code>max(dist)</code> .
type	Diversity type(s) to compute. "auto" (default) returns every type whose inputs are present (always neutral, plus phylogenetic with a tree and functional with a dist). Pass an explicit type, or a character vector of types, to restrict the output – e.g. "neutral" ignores any tree/dist carried by the object, and <code>c("neutral", "phylogenetic")</code> drops functional even when a dist is supplied. A requested type that lacks its input (e.g. "phylogenetic" without a tree) is an error.
reference	Reference tree depth for <i>phylogenetic</i> Hill numbers (ignored for neutral and functional types). "pool" (default) reads every sample at one common depth $T = \text{mean}(T_j)$, so values share a comparable axis across samples; "sample" reads each sample at its own depth T_j (effective lineages at that sample's depth). The two coincide on ultrametric trees. This reference depth is intentionally <i>not</i> offered by <code>hillpart()</code> : in a partition T is fixed at the mean per-sample depth of Chiu et al. (2014), the unique value for which γ / α is a valid decomposition with β in $[1, N]$.
out	Output shape: "tibble" (default) returns a long-format <code>data.frame</code> with columns <code>q</code> , <code>sample</code> , <code>value</code> (plus a <code>type</code> column when more than one type is computed) and <code>print()/plot()</code> methods; "matrix" returns a matrix with samples in rows and diversity orders (<code>q0</code> , <code>q1</code> , ...) in columns, or, when more than one type is computed, a named list of such matrices (one per type).

Value

A long-format `data.frame` of class `hill_diversity` (default). With `out = "matrix"`, a matrix of Hill numbers (samples in rows, diversity orders `q0`, `q1`, ... in columns) for a single type, or a named list of such matrices when several types are computed.

References

Chao, A., Chiu, C.-H. & Jost, L. (2010). Phylogenetic diversity measures based on Hill numbers. *Phil. Trans. R. Soc. B*, 365, 3599-3609.

Alberdi, A. & Gilbert, M.T.P. (2019). A guide to the application of Hill numbers to DNA-based diversity analyses. *Mol. Ecol. Resour.*, 19, 804-817.

See Also

[hillpart\(\)](#), [hilldiss\(\)](#), [hillprof\(\)](#)

Examples

```
counts <- matrix(c(10, 0, 5, 2, 8, 1), nrow = 3,
                 dimnames = list(c("t1", "t2", "t3"), c("s1", "s2")))
hilldiv(counts)
hilldiv(counts, q = c(0, 1, 2))
plot(hilldiv(counts, q = c(0, 1, 2)))

# Supplying both a tree and a distance matrix returns neutral, phylogenetic
# and functional diversity together, distinguished by a `type` column.
tree <- ape::read.tree(text = "((t1:1,t2:1):1,t3:2);")
dist <- as.matrix(stats::dist(c(t1 = 0, t2 = 1, t3 = 4)))
hilldiv(counts, tree = tree, dist = dist)

# Restrict the output with `type` (a scalar or a vector):
hilldiv(counts, tree = tree, dist = dist, type = c("neutral", "functional"))
```

hilleven

Hill-number evenness

Description

Evenness expressed through Hill numbers as the ratio of diversity of order q to richness ($qD / \emptyset D$), which ranges from 0 to 1.

Usage

```
hilleven(
  data,
  q = c(1, 2),
  tree = NULL,
  dist = NULL,
  tau = NULL,
  type = c("auto", "neutral", "phylogenetic", "functional"),
  out = c("tibble", "matrix")
)
```

Arguments

data	Counts: a numeric vector (one sample), a matrix/data.frame (taxa x samples), a phyloseq object or a TreeSummarizedExperiment.
q	Numeric vector of diversity orders (> 0 are meaningful for evenness). Defaults to c(1, 2).
tree	A phylogenetic tree of class phylo whose tip labels match the taxa in data.
dist	A functional distance matrix (or dist) over the taxa.
tau	Optional functional distance threshold. Defaults to max(dist).
type	Diversity type: "auto" (default) infers it from the inputs (counts only -> neutral, +tree -> phylogenetic, +dist -> functional); an explicit "neutral", "phylogenetic" or "functional" asserts the type and is validated against the inputs (e.g. "phylogenetic" requires a tree; "neutral" ignores any tree/dist carried by the object).
out	Output shape: "tibble" (default) returns a long-format data.frame with columns q, sample, value; "matrix" returns a matrix with samples in rows and orders in columns.

Value

A long-format data.frame of class hill_evenness (default) with a plot() method, or a matrix of evenness values (samples in rows, orders in columns) when out = "matrix".

See Also

`hilldiv()`

Examples

```
counts <- matrix(c(10, 0, 5, 2, 8, 1), nrow = 3,
                 dimnames = list(c("t1", "t2", "t3"), c("s1", "s2")))
hilleven(counts)
plot(hilleven(counts, q = c(1, 1.5, 2)))
```

hillpair

Pairwise Hill numbers-based dissimilarity

Description

Compute dissimilarity metrics for every pair of samples, returning distance objects suitable for ordination (e.g. NMDS, PCoA).

Usage

```
hillpair(
  data,
  q = c(0, 1, 2),
  metric = c("S", "C", "U", "V"),
  tree = NULL,
  dist = NULL,
  tau = NULL,
  type = c("auto", "neutral", "phylogenetic", "functional"),
  out = c("dist", "tibble"),
  parallel = FALSE
)
```

Arguments

data	A count table (taxa x samples) or a supported object; a single sample is not meaningful for partitioning.
q	Numeric vector of diversity orders (≥ 0). Defaults to <code>c(0, 1, 2)</code> (richness, Shannon, Simpson).
metric	Dissimilarity metric(s) to return, any of "S", "C", "U", "V". Defaults to all four.
tree	A phylogenetic tree of class <code>phylo</code> whose tip labels match the taxa in data.
dist	A functional distance matrix (or <code>dist</code>) over the taxa.
tau	Optional functional distance threshold. Defaults to <code>max(dist)</code> .
type	Diversity type: "auto" (default) infers it from the inputs (counts only -> neutral, +tree -> phylogenetic, +dist -> functional); an explicit "neutral", "phylogenetic" or "functional" asserts the type and is validated against the inputs (e.g. "phylogenetic" requires a tree; "neutral" ignores any tree/dist carried by the object).
out	Output type: "dist" (default) returns a <code>dist</code> object per requested metric/order combination; "tibble" returns a long-format table.
parallel	Logical; if TRUE and <code>furrr</code> is installed, compute pairs in parallel.

Details

The type-specific structure (per-sample normalisation, the tree traversal or the functional similarity product) is computed **once** over all samples via the partitioning engine; each pair then only combines its two precomputed columns into beta, which is turned into the requested overlap metrics. The maths are therefore identical to `hilldiss()` on two samples, without re-running the full engine per pair. When `parallel = TRUE` and the `furrr` package is installed, pairs are computed in parallel via the active future plan. A `progressr` progress bar is reported when that package is installed and a handler is active.

Value

For `out = "dist"`, a named list of `dist` objects (one per order/metric, named e.g. "q0S"), collapsed to a single `dist` when only one combination is requested. For `out = "tibble"`, a long-format `data.frame` with columns `first`, `second`, `q`, `metric`, `value`.

See Also

[hilldiss\(\)](#), [hilldiv\(\)](#)

Examples

```
counts <- matrix(c(10, 0, 5, 2, 8, 1, 3, 4, 0, 6, 2, 7), nrow = 3,
                 dimnames = list(c("t1", "t2", "t3"),
                                c("s1", "s2", "s3", "s4")))
hillpair(counts, q = 1, metric = "C")
```

hillpart

Hill numbers diversity partitioning

Description

Partition neutral, phylogenetic or functional Hill-number diversity into alpha, gamma and beta components across a set of samples. With a hierarchy formula it instead performs *multi-scale* (nested) partitioning, returning one beta per hierarchical level.

Usage

```
hillpart(
  data,
  q = c(0, 1, 2),
  tree = NULL,
  dist = NULL,
  tau = NULL,
  hierarchy = NULL,
  metadata = NULL,
  type = c("auto", "neutral", "phylogenetic", "functional"),
  out = c("tibble", "matrix")
)
```

Arguments

data	A count table (taxa x samples) or a supported object; a single sample is not meaningful for partitioning.
q	Numeric vector of diversity orders (≥ 0). Defaults to <code>c(0, 1, 2)</code> (richness, Shannon, Simpson).
tree	A phylogenetic tree of class <code>phylo</code> whose tip labels match the taxa in data.
dist	A functional distance matrix (or <code>dist</code>) over the taxa.
tau	Optional functional distance threshold. Defaults to <code>max(dist)</code> .

hierarchy	Optional one-sided nesting formula, coarsest to finest, e.g. <code>~ region / site</code> , requesting multi-scale (nested) partitioning instead of the default single-level partition. One beta is returned per hierarchical transition and the chain telescopes exactly: <code>gamma = alpha_finest * prod(beta)</code> . Works for all three diversity types (neutral, phylogenetic, functional); see the partitioning vignette for the shared construction and its assumptions (equal per-sample weighting; one shared tree depth / tau across scales). Grouping variables are resolved against metadata when supplied, otherwise against the calling environment.
metadata	Optional per-sample data.frame supplying the variables named in hierarchy; rows are matched to the count-table columns by name when possible, otherwise by position.
type	Diversity type: "auto" (default) infers it from the inputs (counts only -> neutral, +tree -> phylogenetic, +dist -> functional); an explicit "neutral", "phylogenetic" or "functional" asserts the type and is validated against the inputs (e.g. "phylogenetic" requires a tree; "neutral" ignores any tree/dist carried by the object).
out	Output shape: "tibble" (default) returns a long-format data.frame with columns q, component, value; "matrix" returns the legacy matrix (orders in rows, alpha/gamma/beta in columns). With hierarchy, "tibble" returns one row per (q, scale) and "matrix" returns alpha, one beta_<level> per nesting level, and gamma.

Value

A long-format data.frame of class `hill_partition` (default) with a `plot()` method, or a matrix with columns alpha, gamma, beta and diversity orders in rows when `out = "matrix"`. With hierarchy, a `hill_hierarchy` long-format data.frame (with its own `plot()` method) or the corresponding wide matrix.

See Also

[hilldiv\(\)](#), [hilldiss\(\)](#), [hillsim\(\)](#)

Examples

```
counts <- matrix(c(10, 0, 5, 2, 8, 1), nrow = 3,
                 dimnames = list(c("t1", "t2", "t3"), c("s1", "s2")))
hillpart(counts)
plot(hillpart(counts))

# Multi-scale partitioning across a nested design.
set.seed(1)
tab <- matrix(rpois(12 * 8, 5), nrow = 12,
              dimnames = list(paste0("t", 1:12), paste0("s", 1:8)))
md <- data.frame(region = rep(c("N", "S"), each = 4),
                 site = rep(c("a", "b", "c", "d"), each = 2),
                 row.names = paste0("s", 1:8))
hillpart(tab, hierarchy = ~ region / site, metadata = md)
```

hillprof

*Diversity profile across a range of orders***Description**

Compute a diversity profile: Hill numbers evaluated over a fine sweep of diversity orders q . Profiles are the standard diagnostic for comparing the diversity of assemblages, since the ranking of samples can change with q .

Usage

```
hillprof(
  data,
  q = seq(0, 3, by = 0.1),
  tree = NULL,
  dist = NULL,
  tau = NULL,
  type = c("auto", "neutral", "phylogenetic", "functional"),
  reference = c("pool", "sample"),
  out = c("tibble", "matrix")
)
```

Arguments

data	Counts: a numeric vector (one sample), a matrix/data.frame (taxa x samples), a phyloseq object or a TreeSummarizedExperiment.
q	Numeric vector of diversity orders to evaluate. Defaults to a fine sweep from 0 to 3.
tree	A phylogenetic tree of class phylo whose tip labels match the taxa in data.
dist	A functional distance matrix (or dist) over the taxa.
tau	Optional functional distance threshold. Defaults to <code>max(dist)</code> .
type	Diversity type: "auto" (default) infers it from the inputs (counts only -> neutral, +tree -> phylogenetic, +dist -> functional); an explicit "neutral", "phylogenetic" or "functional" asserts the type and is validated against the inputs (e.g. "phylogenetic" requires a tree; "neutral" ignores any tree/dist carried by the object).
reference	Reference tree depth for <i>phylogenetic</i> Hill numbers (ignored for neutral and functional types). "pool" (default) reads every sample at one common depth $T = \text{mean}(T_j)$, so values share a comparable axis across samples; "sample" reads each sample at its own depth T_j (effective lineages at that sample's depth). The two coincide on ultrametric trees. This reference depth is intentionally <i>not</i> offered by <code>hillpart()</code> : in a partition T is fixed at the mean per-sample depth of Chiu et al. (2014), the unique value for which γ / α is a valid decomposition with β in $[1, N]$.
out	Output type: "tibble" (default, long format ready for plotting) or "matrix".

Value

A long-format data.frame of class `hill_profile` (columns `q`, `sample`, `value`) with a `plot()` method, or a matrix (samples in rows, orders in columns) when `out = "matrix"`.

See Also

`hilldiv()`

Examples

```
counts <- matrix(c(10, 0, 5, 2, 8, 1), nrow = 3,
                 dimnames = list(c("t1", "t2", "t3"), c("s1", "s2")))
prof <- hillprof(counts)
plot(prof)
```

hillred

Hill numbers redundancy

Description

Estimate phylogenetic or functional redundancy by fitting the saturating relationship between neutral diversity and phylogenetic/functional diversity across samples: $y = -a * 2^{(-x / b)} + c$. Redundancy is summarised as $1 - b / \max(x)$.

Usage

```
hillred(
  data,
  q = c(0, 1, 2),
  tree = NULL,
  dist = NULL,
  tau = NULL,
  type = c("auto", "phylogenetic", "functional"),
  reference = c("pool", "sample"),
  out = c("tibble", "matrix")
)
```

Arguments

<code>data</code>	A count table (taxa x samples); requires either <code>tree</code> or <code>dist</code> .
<code>q</code>	Numeric vector of diversity orders (≥ 0). Defaults to <code>c(0, 1, 2)</code> (richness, Shannon, Simpson).
<code>tree</code>	A phylogenetic tree of class <code>phylo</code> whose tip labels match the taxa in <code>data</code> .
<code>dist</code>	A functional distance matrix (or <code>dist</code>) over the taxa.
<code>tau</code>	Optional functional distance threshold. Defaults to <code>max(dist)</code> .

type	Diversity type: "auto" (default) infers it from the inputs (counts only -> neutral, +tree -> phylogenetic, +dist -> functional); an explicit "neutral", "phylogenetic" or "functional" asserts the type and is validated against the inputs (e.g. "phylogenetic" requires a tree; "neutral" ignores any tree/dist carried by the object).
reference	Reference tree depth for <i>phylogenetic</i> Hill numbers (ignored for neutral and functional types). "pool" (default) reads every sample at one common depth $T = \text{mean}(T_j)$, so values share a comparable axis across samples; "sample" reads each sample at its own depth T_j (effective lineages at that sample's depth). The two coincide on ultrametric trees. This reference depth is intentionally <i>not</i> offered by <code>hillpart()</code> : in a partition T is fixed at the mean per-sample depth of Chiu et al. (2014), the unique value for which γ / α is a valid decomposition with β in $[1, N]$.
out	Output shape: "tibble" (default) returns a <code>data.frame</code> with one row per q and columns q , redundancy, a , b , c ; "matrix" returns the legacy matrix (orders in rows).

Value

A `data.frame` of class `hill_redundancy` (default) with a `plot()` method, or a matrix with columns redundancy, a , b , c (one row per q) when `out = "matrix"`. The tibble carries the per-sample neutral and phylogenetic/functional diversity used for the fit as a "hill_fit" attribute, which the plot method draws.

See Also

`hilldiv()`, `plot.hill_redundancy()`

Examples

```
d <- traits2dist(gut_traits)
red <- hillred(gut_counts, dist = d)
red
plot(red)
```

hillsim

Hill numbers-based similarity

Description

Compute overall similarity metrics from the Hill-number beta diversity (Chiu et al. 2014). These are 1 - the dissimilarities from `hilldiss()`.

Usage

```
hillsim(
  data,
  q = c(0, 1, 2),
  metric = c("S", "C", "U", "V"),
  tree = NULL,
  dist = NULL,
  tau = NULL,
  type = c("auto", "neutral", "phylogenetic", "functional"),
  out = c("tibble", "matrix")
)
```

Arguments

<code>data</code>	A count table (taxa x samples) or a supported object; a single sample is not meaningful for partitioning.
<code>q</code>	Numeric vector of diversity orders (≥ 0). Defaults to <code>c(0, 1, 2)</code> (richness, Shannon, Simpson).
<code>metric</code>	Dissimilarity metric(s) to return, any of "S", "C", "U", "V". Defaults to all four.
<code>tree</code>	A phylogenetic tree of class <code>phylo</code> whose tip labels match the taxa in <code>data</code> .
<code>dist</code>	A functional distance matrix (or <code>dist</code>) over the taxa.
<code>tau</code>	Optional functional distance threshold. Defaults to <code>max(dist)</code> .
<code>type</code>	Diversity type: "auto" (default) infers it from the inputs (counts only -> neutral, +tree -> phylogenetic, +dist -> functional); an explicit "neutral", "phylogenetic" or "functional" asserts the type and is validated against the inputs (e.g. "phylogenetic" requires a tree; "neutral" ignores any tree/dist carried by the object).
<code>out</code>	Output shape: "tibble" (default) returns a long-format data.frame with columns <code>q</code> , <code>metric</code> , <code>value</code> ; "matrix" returns the legacy matrix (orders in rows, metrics in columns, dropped to a vector for a single metric).

Value

A long-format data.frame of class `hill_similarity` (default, with a `plot()` method), or a matrix/vector of similarities when `out = "matrix"`.

See Also

[hilldiss\(\)](#)

Examples

```
counts <- matrix(c(10, 0, 5, 2, 8, 1), nrow = 3,
  dimnames = list(c("t1", "t2", "t3"), c("s1", "s2")))
hillsim(counts)
plot(hillsim(counts))
```

match_data	<i>Match and align a count table to a tree or distance matrix</i>
------------	---

Description

Subsets and reorders a count table so that its taxa match those of a phylogenetic tree or a functional distance matrix, dropping taxa absent from the reference. This realises the `match_data()` helper that `hilldiv2`'s documentation referred to but never provided.

Usage

```
match_data(data, tree = NULL, dist = NULL)
```

Arguments

<code>data</code>	A count matrix/data.frame (taxa x samples) with row names.
<code>tree</code>	A phylo tree (optional).
<code>dist</code>	A distance matrix (optional).

Value

The count matrix restricted to and ordered by the shared taxa.

Examples

```
counts <- matrix(1:6, nrow = 3,
  dimnames = list(c("t1", "t2", "t3"), c("s1", "s2")))
tree <- ape::read.tree(text = "((t1:1,t2:1):1,t4:2);")
match_data(counts, tree = tree)
```

plot.hill_profile	<i>Plot a diversity profile</i>
-------------------	---------------------------------

Description

Base-graphics plot of a `hillprof()` result: one line per sample showing the Hill number against the diversity order q .

Usage

```
## S3 method for class 'hill_profile'
plot(x, ...)
```

Arguments

<code>x</code>	A <code>hill_profile</code> object from <code>hillprof()</code> .
<code>...</code>	Further arguments passed to <code>plot()</code> .

Value

The hill_profile object, invisibly.

plot.hill_redundancy *Plot a redundancy fit*

Description

Base-graphics plot of a [hillred\(\)](#) result. For each diversity order q it shows the per-sample neutral diversity (x) against phylogenetic/functional diversity (y), overlaid with the fitted saturating curve $y = -a * 2^{(-x / b)} + c$. A curve that bends sharply and plateaus well below the points' spread indicates high redundancy; a near-linear fit indicates low redundancy. This mirrors the profile plot of [hillprof\(\)](#).

Usage

```
## S3 method for class 'hill_redundancy'
plot(x, ...)
```

Arguments

x A hill_redundancy object from [hillred\(\)](#).
 ... Further arguments passed to [plot\(\)](#).

Value

The hill_redundancy object, invisibly.

See Also

[hillred\(\)](#), [plot.hill_profile\(\)](#)

traits2dist *Convert a trait table into a distance matrix*

Description

Build a pairwise functional distance matrix from a table of taxon traits, suitable as the dist argument of [hilldiv\(\)](#) and friends.

Usage

```
traits2dist(traits, method = c("gower", "euclidean", "manhattan"))
```

Arguments

traits	A table with taxa (OTUs/ASVs/MAGs) in rows and traits in columns. Traits may be continuous, binary or proportional.
method	Distance metric passed to <code>cluster::daisy()</code> : "gower" (default), "euclidean" or "manhattan".

Value

A numeric distance matrix.

Examples

```
traits <- data.frame(body = c(1, 0.2, 0.9), diet = c(0L, 1L, 1L),
  row.names = c("t1", "t2", "t3"))
traits2dist(traits)
```

tss	<i>Total Sum Scaling normalisation</i>
-----	--

Description

Normalise a numeric vector or count matrix so that each sample (column) sums to one. Columns that sum to zero are returned as all-zero (the 0/0 = NaN case is mapped to 0).

Usage

```
tss(abund)
```

Arguments

abund	A numeric vector or a matrix/data.frame of counts with taxa (OTUs/ASVs/MAGs) in rows and samples in columns.
-------	--

Value

A normalised object of the same shape as abund (vector in, vector out; matrix/data.frame in, matrix out).

Examples

```
tss(c(a = 1, b = 3))
tss(matrix(c(1, 0, 3, 0, 0, 2), nrow = 3))
```

Index

* datasets

- gut_counts, [2](#)
- gut_traits, [3](#)
- gut_tree, [4](#)

cluster::daisy(), [18](#)

gut_counts, [2](#), [3](#), [4](#)
gut_traits, [3](#), [3](#), [4](#)
gut_tree, [3](#), [4](#)

hilldiss, [4](#)
hilldiss(), [7](#), [9–11](#), [14](#), [15](#)
hilldiv, [5](#)
hilldiv(), [8](#), [10](#), [11](#), [13](#), [14](#), [17](#)
hilleven, [7](#)
hillpair, [8](#)
hillpair(), [5](#)
hillpart, [10](#)
hillpart(), [5–7](#), [12](#), [14](#)
hillprof, [12](#)
hillprof(), [7](#), [16](#), [17](#)
hillred, [13](#)
hillred(), [17](#)
hillsim, [14](#)
hillsim(), [4](#), [5](#), [11](#)

match_data, [16](#)

plot(), [13](#), [14](#), [16](#), [17](#)
plot.hill_profile, [16](#)
plot.hill_profile(), [17](#)
plot.hill_redundancy, [17](#)
plot.hill_redundancy(), [14](#)

traits2dist, [17](#)
traits2dist(), [3](#)
tss, [18](#)