Package 'divent'

Type Package

Title Entropy Partitioning to Measure Diversity

Version 0.4-4

Description Measurement and partitioning of diversity, based on Tsallis entropy, following Marcon and Herault (2015) <doi:10.18637/jss.v067.i08>.
'divent' provides functions to estimate alpha, beta and gamma diversity of communities, including phylogenetic and functional diversity.

URL https://ericmarcon.github.io/divent/,

https://github.com/EricMarcon/divent

BugReports https://github.com/EricMarcon/divent/issues

License GNU General Public License

Depends R (>= 3.5)

Imports ape, cli, dplyr, ggplot2, graphics, EntropyEstimation,
RColorBrewer, Rdpack, rlang, stats, tibble, tidyr, vegan

Suggests ade4, knitr, pkgdown, rmarkdown, SPECIES, testthat (>= 3.0.0)

Encoding UTF-8

LazyData true

VignetteBuilder knitr

SystemRequirements pandoc

RdMacros Rdpack

RoxygenNote 7.3.2

Config/testthat/edition 3

NeedsCompilation no

Author Eric Marcon [aut, cre] (<https://orcid.org/0000-0002-5249-321X>)

Maintainer Eric Marcon <eric.marcon@agroparistech.fr>

Repository CRAN

Date/Publication 2024-11-06 16:10:08 UTC

Contents

`divent-package`   *divent*

Description

Measures of Diversity and Entropy

Details

This package is a reboot of the **entropart** package (Marcon and Hérault 2015).

Author(s)

Maintainer: Eric Marcon <eric.marcon@agroparistech.fr> ([ORCID](#))

References

Marcon E, Hérault B (2015). "Entropart, an R Package to Measure and Partition Diversity." *Journal of Statistical Software*, **67**(8), 1–26. [doi:10.18637/jss.v067.i08](#).

See Also

Useful links:

- <https://ericmarcon.github.io/divent/>
- <https://github.com/EricMarcon/divent>
- Report bugs at <https://github.com/EricMarcon/divent/issues>

`abd_freq_count`   *Abundance Frequency Count of a Community*

Description

Count the number of species observed the same number of times.

Usage

```
abd_freq_count(
  abd,
  level = NULL,
  probability_estimator = c("naive", "Chao2013", "Chao2015", "ChaoShen"),
  unveiling = c("none", "uniform", "geometric"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "rarefy", "naive"),
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  check_arguments = TRUE
)
```

Arguments

abd	A numeric vector containing species abundances.
level	The level of interpolation or extrapolation. It may be a sample size (an integer) or a sample coverage (a number between 0 and 1). If not NULL, the asymptotic estimator is ignored.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see probabilities). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see probabilities). Used only for extrapolation.
richness_estimator	
	A string containing an estimator recognized by div_richness to evaluate the total number of species in probabilities. Used only for extrapolation.
coverage_estimator	
	An estimator of sample coverage used by coverage.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

Details

The Abundance Frequency Count (Chao and Jost 2015) is the number of species observed each number of times. It is a way to summarize the species distribution.

It can be estimated at a specified level of interpolation or extrapolation. Extrapolation relies on the estimation of the estimation of the asymptotic distribution of the community by probabilities and eq. (5) of (Chao et al. 2014).

Value

A two-column tibble. The first column contains the number of observations, the second one the number of species observed this number of times.

Examples

```
abd_freq_count(paracou_6_abd[1, -(1:2)])
```

---

abd_species	*Abundances of Communities*

Description

Utilities for community abundances (objects of class "abundances").

Usage

```
abd_species(abundances, check_arguments = TRUE)

abd_sum(abundances, as_numeric = FALSE, check_arguments = TRUE)

prob_species(species_distribution, check_arguments = TRUE)
```

Arguments

abundances   An object of class [abundances](#).

check_arguments

If TRUE, the function arguments are verified. Should be set to FALSE to save time
when the arguments have been checked elsewhere.

as_numeric   If TRUE, a number or a numeric vector is returned rather than a tibble.

species_distribution

An object of class [species_distribution](#).

Value

abd_species() returns a tibble containing the species abundance columns only, to simplify numeric operations.

prob_species() returns the same tibble but values are probabilities.

abd_sum() returns the sample sizes of the communities in a numeric vector.

Examples

```
abd_species(paracou_6_abd)
abd_sum(paracou_6_abd)
```

---

accum_div_phylo   *Phylogenetic Diversity Accumulation of a Community*

Description

Diversity and Entropy Accumulation Curves represent the accumulation of entropy with respect to
the sample size.

Usage

```
accum_ent_phylo(x, ...)

## S3 method for class 'numeric'
accum_ent_phylo(
  x,
  tree,
```

```
  q = 0,
  normalize = TRUE,
  levels = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("rarefy", "jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  n_simulations = 0,
  alpha = 0.05,
  show_progress = TRUE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'abundances'
accum_ent_phylo(
  x,
  tree,
  q = 0,
  normalize = TRUE,
  levels = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("rarefy", "jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  gamma = FALSE,
  n_simulations = 0,
  alpha = 0.05,
  show_progress = TRUE,
  ...,
  check_arguments = TRUE
)

accum_div_phylo(x, ...)

## S3 method for class 'numeric'
accum_div_phylo(
  x,
  tree,
  q = 0,
  normalize = TRUE,
  levels = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
```

```
    richness_estimator = c("rarefy", "jackknife", "iChao1", "Chao1", "naive"),
    jack_alpha = 0.05,
    jack_max = 10,
    coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
    n_simulations = 0,
    alpha = 0.05,
    show_progress = TRUE,
    ...,
    check_arguments = TRUE
)

## S3 method for class 'abundances'
accum_div_phylo(
  x,
  tree,
  q = 0,
  normalize = TRUE,
  levels = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("rarefy", "jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  gamma = FALSE,
  n_simulations = 0,
  alpha = 0.05,
  show_progress = TRUE,
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities.
...	Unused.
tree	An ultrametric, phylogenetic tree. May be an object of class phylo_divent, ape::phylo, ade4::phylog or stats::hclust.
q	The order of diversity.
normalize	If TRUE, phylogenetic is normalized: the height of the tree is set to 1.
levels	The levels, i.e. the sample sizes of interpolation or extrapolation: a vector of integer values.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see probabilities). Used only for extrapolation.

unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see probabilities). Used only for extrapolation.
richness_estimator	
	An estimator of richness to evaluate the total number of species, see div_richness. Used for interpolation and extrapolation.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by coverage.
n_simulations	The number of simulations used to estimate the confidence envelope.
alpha	The risk level, 5% by default.
show_progress	If TRUE, a progress bar is shown during long computations.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
gamma	If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

accum_ent_phylo() or accum_div_phylo() estimate the phylogenetic diversity or entropy accumulation curve of a distribution. See ent_tsallis for details about the computation of entropy at each level of interpolation and extrapolation.

In accumulation curves, extrapolation if done by estimating the asymptotic distribution of the community and estimating entropy at different levels by interpolation.

Interpolation and extrapolation of integer orders of diversity are from Chao et al. (2014). The asymptotic richness is adjusted so that the extrapolated part of the accumulation joins the observed value at the sample size.

"accumulation" objects can be plotted. They generalize the classical Species Accumulation Curves (SAC) which are diversity accumulation of order $q = 0$.

Value

A tibble with the site names, the estimators used and the accumulated entropy or diversity at each level of sampling effort.

References

Chao A, Gotelli NJ, Hsieh TC, Sander EL, Ma KH, Colwell RK, Ellison AM (2014). "Rarefaction and Extrapolation with Hill Numbers: A Framework for Sampling and Estimation in Species Diversity Studies." *Ecological Monographs*, **84**(1), 45–67. doi:10.1890/130133.1.

Examples

```
# Richness accumulation up to the sample size.
# 100 simulations only to save time.
autoplot(
  accum_div_phylo(mock_3sp_abd, tree = mock_3sp_tree, n_simulations = 100)
)
```

---

accum_hill *Diversity Accumulation of a Community*

Description

Diversity and Entropy Accumulation Curves represent the accumulation of entropy with respect to the sample size.

Usage

```
accum_tsallis(x, ...)

## S3 method for class 'numeric'
accum_tsallis(
  x,
  q = 0,
  levels = seq_len(sum(x)),
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("rarefy", "jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  n_simulations = 0,
  alpha = 0.05,
  show_progress = TRUE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'abundances'
accum_tsallis(
  x,
  q = 0,
  levels = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("rarefy", "jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
```

```
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  n_simulations = 0,
  alpha = 0.05,
  show_progress = TRUE,
  ...,
  check_arguments = TRUE
)

accum_hill(x, ...)

## S3 method for class 'numeric'
accum_hill(
  x,
  q = 0,
  levels = seq_len(sum(x)),
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("rarefy", "jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  n_simulations = 0,
  alpha = 0.05,
  show_progress = TRUE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'abundances'
accum_hill(
  x,
  q = 0,
  levels = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("rarefy", "jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  n_simulations = 0,
  alpha = 0.05,
  show_progress = TRUE,
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities.
...	Unused.
q	The order of diversity.
levels	The levels, i.e. the sample sizes of interpolation or extrapolation: a vector of integer values.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see probabilities). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see probabilities). Used only for extrapolation.
richness_estimator	
	An estimator of richness to evaluate the total number of species, see div_richness. Used for interpolation and extrapolation.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by coverage.
n_simulations	The number of simulations used to estimate the confidence envelope.
alpha	The risk level, 5% by default.
show_progress	If TRUE, a progress bar is shown during long computations.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

Details

accum_hill() or accum_tsallis() estimate the diversity or entropy accumulation curve of a distribution. See ent_tsallis for details about the computation of entropy at each level of interpolation and extrapolation.

In accumulation curves, extrapolation is done by estimating the asymptotic distribution of the community and estimating entropy at different levels by interpolation.

Interpolation and extrapolation of integer orders of diversity are from Chao et al. (2014). The asymptotic richness is adjusted so that the extrapolated part of the accumulation joins the observed value at the sample size.

"accumulation" objects can be plotted. They generalize the classical Species Accumulation Curves (SAC) which are diversity accumulation of order $q = 0$.

Value

A tibble with the site names, the estimators used and the accumulated entropy or diversity at each level of sampling effort.

References

Chao A, Gotelli NJ, Hsieh TC, Sander EL, Ma KH, Colwell RK, Ellison AM (2014). "Rarefaction and Extrapolation with Hill Numbers: A Framework for Sampling and Estimation in Species Diversity Studies." *Ecological Monographs*, **84**(1), 45–67. doi:10.1890/130133.1.

Examples

```
# Paracou 6 subplot 1
autoplot(accum_hill(paracou_6_abd[1, ]))
```

---

autoplot.accumulation   *Plot Accumulation Objects*

Description

Plot objects of class "accumulation" produced by accum_hill and other accumulation functions.

Usage

```
## S3 method for class 'accumulation'
autoplot(
  object,
  ...,
  main = NULL,
  xlab = "Sample Size",
  ylab = NULL,
  shade_color = "grey75",
  alpha = 0.3,
  lty = ggplot2::GeomLine$default_aes$linetype,
  lwd = ggplot2::GeomLine$default_aes$linewidth
)
```

Arguments

object	An object of class "accumulation".
...	Unused.
main	The main title of the plot.
xlab	The label of the x-axis.
ylab	The label of the y-axis.
shade_color	The color of the shaded confidence envelopes.
alpha	The opacity of the confidence envelopes, between 0 (transparent) and 1 (opaque).
lty	The line type of the curves.
lwd	The line width of the curves.

Value

A [ggplot2::ggplot](#) object.

Examples

```
# Species accumulation curve
autoplot(accum_hill(mock_3sp_abd))
```

---

autoplot.profile   *Plot Profile Objects*

Description

Plot objects of class "profile" produced by [profile_hill](#) and other profile functions.

Usage

```
## S3 method for class 'profile'
autoplot(
  object,
  ...,
  main = NULL,
  xlab = "Order of Diversity",
  ylab = "Diversity",
  shade_color = "grey75",
  alpha = 0.3,
  lty = ggplot2::GeomLine$default_aes$linetype,
  lwd = ggplot2::GeomLine$default_aes$linewidth
)
```

Arguments

object	An object of class "profile".
...	Unused.
main	The main title of the plot.
xlab	The label of the x-axis.
ylab	The label of the y-axis.
shade_color	The color of the shaded confidence envelopes.
alpha	The opacity of the confidence envelopes, between 0 (transparent) and 1 (opaque).
lty	The line type of the curves.
lwd	The line width of the curves.

Value

A [ggplot2::ggplot](#) object.

Examples

```
# Diversity profile curve
autoplot(profile_hill(mock_3sp_abd))
```

---

coverage   *Sample Coverage of a Community*

Description

coverage() calculates an estimator of the sample coverage of a community described by its abundance vector. coverage_to_size() estimates the sample size corresponding to the chosen sample coverage.

Usage

```
coverage(x, ...)

## S3 method for class 'numeric'
coverage(
  x,
  estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  level = NULL,
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'abundances'
coverage(
  x,
  estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  level = NULL,
  ...,
  check_arguments = TRUE
)

coverage_to_size(x, ...)

## S3 method for class 'numeric'
coverage_to_size(
  x,
  sample_coverage,
  estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
```

```
)

## S3 method for class 'abundances'
coverage_to_size(
  x,
  sample_coverage,
  estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object.
...	Unused.
estimator	An estimator of the sample coverage. "ZhangHuang" is the most accurate but does not allow choosing a `level`. "Good"'s estimator only allows interpolation, i.e. estimation of the coverage of a subsample. "Chao" allows estimation at any `level`, including extrapolation. "Turing" is the simplest estimator, equal to 1 minus the number of singletons divided by the sample size.
level	The level of interpolation or extrapolation, i.e. an abundance.
as_numeric	If `TRUE`, a number or a numeric vector is returned rather than a tibble.
check_arguments	If `TRUE`, the function arguments are verified. Should be set to `FALSE` to save time when the arguments have been checked elsewhere.
sample_coverage	The target sample coverage.

Details

The sample coverage C of a community is the total probability of occurrence of the species observed in the sample. $1 - C$ is the probability for an individual of the whole community to belong to a species that has not been sampled.

The historical estimator is due to Turing (Good 1953). It only relies on singletons (species observed only once). Chao's (Chao and Shen 2010) estimator uses doubletons too and Zhang-Huang's (Chao et al. 1988; Zhang and Huang 2007) uses the whole distribution.

If `level` is not `NULL`, the sample coverage is interpolated or extrapolated. Interpolation by the Good estimator relies on the equality between sampling deficit and the generalized Simpson entropy (Good 1953). The Chao et al. (2014) estimator allows extrapolation, reliable up a level equal to the double size of the sample.

Value

`coverage()` returns a named number equal to the calculated sample coverage. The name is that of the estimator used.

`coverage_to_size()` returns a number equal to the sample size corresponding to the chosen sample coverage.

References

Chao A, Gotelli NJ, Hsieh TC, Sander EL, Ma KH, Colwell RK, Ellison AM (2014). "Rarefaction and Extrapolation with Hill Numbers: A Framework for Sampling and Estimation in Species Diversity Studies." *Ecological Monographs*, **84**(1), 45–67. doi:10.1890/130133.1.

Chao A, Lee S, Chen T (1988). "A Generalized Good's Nonparametric Coverage Estimator." *Chinese Journal of Mathematics*, **16**, 189–199. 43836340.

Chao A, Shen T (2010). *Program SPADE: Species Prediction and Diversity Estimation. Program and User's Guide.*. CARE.

Good IJ (1953). "The Population Frequency of Species and the Estimation of Population Parameters." *Biometrika*, **40**(3/4), 237–264. doi:10.1093/biomet/40.34.237.

Zhang Z, Huang H (2007). "Turing's Formula Revisited." *Journal of Quantitative Linguistics*, **14**(2-3), 222–241. doi:10.1080/09296170701514189.

Examples

```
coverage(paracou_6_abd)
coverage_to_size(paracou_6_abd, sample_coverage = 0.9)
```

---

div_hill   *Hill number of a Community*

Description

Estimate the diversity sensu stricto, i.e. the Hill (1973) number of species from abundance or probability data.

Usage

```
div_hill(x, q = 1, ...)

## S3 method for class 'numeric'
div_hill(
  x,
  q = 1,
 estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Marcon",
    "UnveilC", "UnveiliC", "ZhangGrabchak", "naive", "Bonachela", "Holste"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
```

```
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  sample_coverage = NULL,
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
div_hill(
  x,
  q = 1,
  estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Marcon",
    "UnveilC", "UnveiliC", "ZhangGrabchak", "naive", "Bonachela", "Holste"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  gamma = FALSE,
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class [abundances](#) or [probabilities](#).
q	The order of diversity.
...	Unused.
estimator	An estimator of asymptotic diversity.
level	The level of interpolation or extrapolation. It may be a sample size (an integer) or a sample coverage (a number between 0 and 1). If not NULL, the asymptotic estimator is ignored.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see [probabilities](#)). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see [probabilities](#)). Used only for extrapolation.
richness_estimator	
	An estimator of richness to evaluate the total number of species, see [div_richness](#). Used for interpolation and extrapolation.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.

coverage_estimator

        An estimator of sample coverage used by coverage.

sample_coverage

        The sample coverage of x calculated elsewhere. Used to calculate the gamma diversity of meta-communities, see details.

as_numeric   If TRUE, a number or a numeric vector is returned rather than a tibble.

check_arguments

        If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

gamma   If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

Several estimators are available to deal with incomplete sampling.

Bias correction requires the number of individuals.

Estimation techniques are from Chao and Shen (2003), Grassberger (1988),Holste et al. (1998), Bonachela et al. (2008), Marcon et al. (2014) which is actually the max value of "ChaoShen" and "Grassberger", Zhang and Grabchak (2014), Chao et al. (2015), Chao and Jost (2015) and Marcon (2015).

The ChaoJost estimator Chao et al. (2013); Chao and Jost (2015) contains an unbiased part concerning observed species, equal to that of Zhang and Grabchak (2014), and a (biased) estimator of the remaining bias based on the estimation of the species-accumulation curve. It is very efficient but slow if the number of individuals is more than a few hundreds.

The unveiled estimators rely on Chao et al. (2015), completed by Marcon (2015). The actual probabilities of observed species are estimated and completed by a geometric distribution of the probabilities of unobserved species. The number of unobserved species is estimated by the Chao1 estimator (UnveilC), following Chao et al. (2015), or by the iChao1 (UnveiliC) or the jackknife (UnveilJ). The UnveilJ correction often has a lower bias but a greater variance (Marcon 2015). It is a good first choice thanks to the versatility of the jackknife estimator of richness.

Estimators by Bonachela et al. (2008) and Holste et al. (1998) are rarely used.

To estimate γ diversity, the size of a metacommunity (see metacommunity) is unknown so it has to be set according to a rule which does not ensure that its abundances are integer values. Then, classical bias-correction methods do not apply. Providing the sample_coverage argument allows applying the ChaoShen and Grassberger corrections to estimate quite well the entropy.

Diversity can be estimated at a specified level of interpolation or extrapolation, either a chosen sample size or sample coverage (Chao et al. 2014), rather than its asymptotic value. See accum_hill for details.

Value

A tibble with the site names, the estimators used and the estimated diversity.

References

Bonachela JA, Hinrichsen H, Muñoz MA (2008). "Entropy Estimates of Small Data Sets." *Journal of Physics A: Mathematical and Theoretical*, **41**(202001), 1–9. doi:10.1088/17518113/41/20/

202001.

Chao A, Gotelli NJ, Hsieh TC, Sander EL, Ma KH, Colwell RK, Ellison AM (2014). "Rarefaction and Extrapolation with Hill Numbers: A Framework for Sampling and Estimation in Species Diversity Studies." *Ecological Monographs*, **84**(1), 45–67. doi:10.1890/130133.1.

Chao A, Hsieh TC, Chazdon RL, Colwell RK, Gotelli NJ (2015). "Unveiling the Species-Rank Abundance Distribution by Generalizing Good-Turing Sample Coverage Theory." *Ecology*, **96**(5), 1189–1201. doi:10.1890/140550.1.

Chao A, Jost L (2015). "Estimating Diversity and Entropy Profiles via Discovery Rates of New Species." *Methods in Ecology and Evolution*, **6**(8), 873–882. doi:10.1111/2041210X.12349.

Chao A, Shen T (2003). "Nonparametric Estimation of Shannon's Index of Diversity When There Are Unseen Species in Sample." *Environmental and Ecological Statistics*, **10**(4), 429–443. doi:10.1023/A:1026096204727.

Chao A, Wang Y, Jost L (2013). "Entropy and the Species Accumulation Curve: A Novel Entropy Estimator via Discovery Rates of New Species." *Methods in Ecology and Evolution*, **4**(11), 1091–1100. doi:10.1111/2041210x.12108.

Grassberger P (1988). "Finite Sample Corrections to Entropy and Dimension Estimates." *Physics Letters A*, **128**(6-7), 369–373. doi:10.1016/03759601(88)901934.

Hill MO (1973). "Diversity and Evenness: A Unifying Notation and Its Consequences." *Ecology*, **54**(2), 427–432. doi:10.2307/1934352.

Holste D, Große I, Herzel H (1998). "Bayes' Estimators of Generalized Entropies." *Journal of Physics A: Mathematical and General*, **31**(11), 2551–2566.

Marcon E (2015). "Practical Estimation of Diversity from Abundance Data." *HAL*, **01212435**(version 2).

Marcon E, Scotti I, Hérault B, Rossi V, Lang G (2014). "Generalization of the Partitioning of Shannon Diversity." *Plos One*, **9**(3), e90289. doi:10.1371/journal.pone.0090289.

Zhang Z, Grabchak M (2014). "Nonparametric Estimation of Kullback-Leibler Divergence." *Neural computation*, **26**(11), 2570–2593. doi:10.1162/NECO_a_00646, 25058703.

Examples

```
# Diversity of each community
div_hill(paracou_6_abd, q = 2)
# gamma diversity
div_hill(paracou_6_abd, q = 2, gamma = TRUE)

# At 80% coverage
div_hill(paracou_6_abd, q = 2, level = 0.8)
```

---

div_hurlbert   *Hurlbert Diversity of a Community*

Description

Estimate the diversity sensu stricto, i.e. the effective number of species number of species Dauby and Hardy (2012) from abundance or probability data.

Usage

```
div_hurlbert(x, k = 1, ...)

## S3 method for class 'numeric'
div_hurlbert(
  x,
  k = 2,
  estimator = c("Hurlbert", "naive"),
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
div_hurlbert(
  x,
  k = 2,
  estimator = c("Hurlbert", "naive"),
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities.
k	The order of Hurlbert's diversity.
...	Unused.
estimator	An estimator of asymptotic diversity.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

Details

Several estimators are available to deal with incomplete sampling.

Bias correction requires the number of individuals.

Estimation techniques are from Hurlbert (1971).

Hurlbert's diversity cannot be estimated at a specified level of interpolation or extrapolation, and diversity partioning is not available.

Value

A tibble with the site names, the estimators used and the estimated diversity.

References

Dauby G, Hardy OJ (2012). "Sampled-Based Estimation of Diversity Sensu Stricto by Transforming Hurlbert Diversities into Effective Number of Species." *Ecography*, **35**(7), 661–672. doi:10.1111/j.16000587.2011.06860.x.

Hurlbert SH (1971). "The Nonconcept of Species Diversity: A Critique and Alternative Parameters." *Ecology*, **52**(4), 577–586. doi:10.2307/1934145.

Examples

```
# Diversity of each community
div_hurlbert(paracou_6_abd, k = 2)
```

---

div_part   *Diversity partition*

Description

Calculate γ, β and α diversities of a metacommunity.

Usage

```
div_part(
  abundances,
  q = 1,
 estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Holste",
    "Marcon", "UnveilC", "UnveiliC", "ZhangGrabchak"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
```

```
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  check_arguments = TRUE
)
```

Arguments

abundances	An object of class abundances.
q	The order of diversity.
estimator	An estimator of diversity.
level	The level of interpolation or extrapolation. It may be a sample size (an integer) or a sample coverage (a number between 0 and 1). If not NULL, the asymptotic estimator is ignored.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see probabilities). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see probabilities). Used only for extrapolation.
richness_estimator	
	An estimator of richness to evaluate the total number of species, see div_richness. Used for interpolation and extrapolation.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by coverage.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

Details

The function computes γ diversity after building a metacommunity from local communities according to their weight (Marcon et al. 2014). α entropy is the weighted mean local entropy, converted into Hill numbers to obtain α diversity. β diversity is obtained as the ratio of γ to α.

Value

A tibble with diversity values at each scale.

References

Marcon E, Scotti I, Hérault B, Rossi V, Lang G (2014). "Generalization of the Partitioning of Shannon Diversity." *Plos One*, **9**(3), e90289. doi:10.1371/journal.pone.0090289.

Examples

```
div_part(paracou_6_abd)
```

div_phylo   *Phylogenetic Diversity of a Community*

Description

Estimate the diversity of species from abundance or probability data and a phylogenetic tree. Several estimators are available to deal with incomplete sampling.

Usage

```
div_phylo(x, tree, q = 1, ...)

## S3 method for class 'numeric'
div_phylo(
  x,
  tree,
  q = 1,
  normalize = TRUE,
 estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Marcon",
    "UnveilC", "UnveiliC", "ZhangGrabchak", "naive", "Bonachela", "Holste"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
div_phylo(
  x,
  tree,
  q = 1,
  normalize = TRUE,
 estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Marcon",
    "UnveilC", "UnveiliC", "ZhangGrabchak", "naive", "Bonachela", "Holste"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
```

```
    gamma = FALSE,
    ...,
    check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities.
tree	An ultrametric, phylogenetic tree. May be an object of class phylo_divent, ape::phylo, ade4::phylog or stats::hclust.
q	The order of diversity.
...	Unused.
normalize	If TRUE, phylogenetic is normalized: the height of the tree is set to 1.
estimator	An estimator of asymptotic diversity.
level	The level of interpolation or extrapolation. It may be a sample size (an integer) or a sample coverage (a number between 0 and 1). If not NULL, the asymptotic estimator is ignored.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see probabilities). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see probabilities). Used only for extrapolation.
richness_estimator	
	An estimator of richness to evaluate the total number of species, see div_richness. Used for interpolation and extrapolation.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by coverage.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
gamma	If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

Bias correction requires the number of individuals. See div_hill for estimators.

Entropy can be estimated at a specified level of interpolation or extrapolation, either a chosen sample size or sample coverage (Chao et al. 2014), rather than its asymptotic value. See accum_tsallis for details.

Value

A tibble with the site names, the estimators used and the estimated diversity

References

Chao A, Gotelli NJ, Hsieh TC, Sander EL, Ma KH, Colwell RK, Ellison AM (2014). "Rarefaction and Extrapolation with Hill Numbers: A Framework for Sampling and Estimation in Species Diversity Studies." *Ecological Monographs*, **84**(1), 45–67. doi:10.1890/130133.1.

Examples

```
div_phylo(paracou_6_abd, tree = paracou_6_taxo, q = 2)

# At 80% coverage
div_phylo(paracou_6_abd, tree = paracou_6_taxo, q = 2, level = 0.8)

# Gamma entropy
div_phylo(paracou_6_abd, tree = paracou_6_taxo, q = 2, gamma = TRUE)
```

---

div_richness   *Number of Species of a Community*

Description

Estimate the number of species from abundance or probability data. Several estimators are available to deal with incomplete sampling.

Usage

```
div_richness(x, ...)

## S3 method for class 'numeric'
div_richness(
  x,
  estimator = c("jackknife", "iChao1", "Chao1", "rarefy", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
)
```

```
## S3 method for class 'species_distribution'
div_richness(
  x,
  estimator = c("jackknife", "iChao1", "Chao1", "rarefy", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  gamma = FALSE,
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities.
...	Unused. The metacommunity if built by combining the community abundances with respect to their weight.
estimator	An estimator of richness to evaluate the total number of species.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
level	The level of interpolation or extrapolation. It may be a sample size (an integer) or a sample coverage (a number between 0 and 1). The asymptotic estimator is used in extrapolation (i.e. a level greater than the sample size).
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see probabilities). Used only by the estimator of richness "rarefy".
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see probabilities). Used only by the estimator of richness "rarefy".
coverage_estimator	
	An estimator of sample coverage used by coverage.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
gamma	If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

Bias correction requires the number of individuals. Chao's correction techniques are from Chao et al. (2014) and Chiu et al. (2014). The Jackknife estimator is calculated by a straight adaptation of the code by Ji-Ping Wang (jackknife in package **SPECIES**). The optimal order is selected according

to Burnham and Overton (1978); Burnham and Overton (1979). Many other estimators are available elsewhere, the ones implemented here are necessary for other entropy estimations.

Richness can be estimated at a specified `level` of interpolation or extrapolation, either a chosen sample size or sample coverage (Chiu et al. 2014), rather than its asymptotic value. Extrapolation relies on the estimation of the asymptotic richness. If `probability_estimator` is "naive", then the asymptotic estimation of richness is made using the chosen `estimator`, else the asymptotic distribution of the community is derived and its estimated richness adjusted so that the richness of a sample of this distribution of the size of the actual sample has the richness of the actual sample.

Value

A tibble with the site names, the estimators used and the estimated numbers of species.

References

Burnham KP, Overton WS (1978). "Estimation of the Size of a Closed Population When Capture Probabilities Vary among Animals." *Biometrika*, **65**(3), 625–633. doi:10.2307/2335915.

Burnham KP, Overton WS (1979). "Robust Estimation of Population Size When Capture Probabilities Vary among Animals." *Ecology*, **60**(5), 927–936. doi:10.2307/1936861.

Chao A, Gotelli NJ, Hsieh TC, Sander EL, Ma KH, Colwell RK, Ellison AM (2014). "Rarefaction and Extrapolation with Hill Numbers: A Framework for Sampling and Estimation in Species Diversity Studies." *Ecological Monographs*, **84**(1), 45–67. doi:10.1890/130133.1.

Chiu C, Wang Y, Walther BA, Chao A (2014). "An Improved Nonparametric Lower Bound of Species Richness via a Modified Good-Turing Frequency Formula." *Biometrics*, **70**(3), 671–682. doi:10.1111/biom.12200, 24945937.

Examples

```
# Diversity of each community
div_richness(paracou_6_abd)
# gamma diversity
div_richness(paracou_6_abd, gamma = TRUE)

# At 80% coverage
div_richness(paracou_6_abd, level = 0.8)
```

---

div_similarity   *Similarity-Based Diversity of a Community*

Description

Estimate the diversity of species from abundance or probability data and a similarity matrix between species. Several estimators are available to deal with incomplete sampling. Bias correction requires the number of individuals.

Usage

```
div_similarity(x, similarities, q = 1, ...)

## S3 method for class 'numeric'
div_similarity(
  x,
  similarities = diag(length(x)),
  q = 1,
  estimator = c("UnveilJ", "Max", "ChaoShen", "MarconZhang", "UnveilC", "UnveiliC",
    "naive"),
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  sample_coverage = NULL,
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
div_similarity(
  x,
  similarities = diag(sum(!colnames(x) %in% non_species_columns)),
  q = 1,
  estimator = c("UnveilJ", "Max", "ChaoShen", "MarconZhang", "UnveilC", "UnveiliC",
    "naive"),
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  gamma = FALSE,
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities. If it is a numeric vector, then its length must equal the dimensions of the similarities matrix: species are assumed to be in the same order.
similarities	A similarity matrix, that can be obtained by fun_similarity. Its default value is the identity matrix.
q	The order of diversity.

...	Unused.
estimator	An estimator of asymptotic diversity.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see [probabilities](#)). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see [probabilities](#)). Used only for extrapolation.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by [coverage](#).
sample_coverage	
	The sample coverage of x calculated elsewhere. Used to calculate the gamma diversity of meta-communities, see details.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
gamma	If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

All species of the `species_distribution` must be found in the matrix of `similarities` if it is named. If it is not, its size must equal the number of species. Then, the order of species is assumed to be the same as that of the `species_distribution`.

Similarity-Based diversity can't be interpolated of extrapolated as of the state of the art.

Value

A tibble with the site names, the estimators used and the estimated diversity.

References

There are no references for Rd macro \insertAllCites on this help page.

Examples

```
# Similarity matrix
Z <- fun_similarity(paracou_6_fundist)
# Diversity of each community
div_similarity(paracou_6_abd, similarities = Z, q = 2)
# gamma diversity
div_similarity(paracou_6_abd, similarities = Z, q = 2, gamma = TRUE)
```

---

ent_hurlbert   *Hurlbert Entropy of a Community*

Description

Estimate the Hurlbert entropy (Hurlbert 1971) of species from abundance or probability data. Several estimators are available to deal with incomplete sampling.

Usage

```
ent_hurlbert(x, k = 2, ...)

## S3 method for class 'numeric'
ent_hurlbert(
  x,
  k = 2,
  estimator = c("Hurlbert", "naive"),
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
ent_hurlbert(
  x,
  k = 2,
  estimator = c("Hurlbert", "naive"),
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities.
k	The order of Hurlbert's diversity.
...	Unused.
estimator	An estimator of entropy.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
check_arguments	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

Details

Bias correction requires the number of individuals. See [div_hurlbert](#) for estimators.

Hurlbert's entropy cannot be estimated at a specified level of interpolation or extrapolation, and entropy partitioning is not available.

Value

A tibble with the site names, the estimators used and the estimated entropy.

References

Hurlbert SH (1971). "The Nonconcept of Species Diversity: A Critique and Alternative Parameters." *Ecology*, **52**(4), 577–586. [doi:10.2307/1934145](https://doi.org/10.2307/1934145).

Examples

```
# Entropy of each community
ent_hurlbert(paracou_6_abd, k = 2)
```

---

ent_phylo   *Phylogenetic Entropy of a Community*

Description

Estimate the entropy of species from abundance or probability data and a phylogenetic tree. Several estimators are available to deal with incomplete sampling.

Usage

```
ent_phylo(x, tree, q = 1, ...)

## S3 method for class 'numeric'
ent_phylo(
  x,
  tree,
  q = 1,
  normalize = TRUE,
  estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Marcon",
    "UnveilC", "UnveiliC", "ZhangGrabchak", "naive", "Bonachela", "Holste"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
```

```
    as_numeric = FALSE,
    ...,
    check_arguments = TRUE
)

## S3 method for class 'species_distribution'
ent_phylo(
    x,
    tree,
    q = 1,
    normalize = TRUE,
  estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Marcon",
      "UnveilC", "UnveiliC", "ZhangGrabchak", "naive", "Bonachela", "Holste"),
    level = NULL,
    probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
    unveiling = c("geometric", "uniform", "none"),
    richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
    jack_alpha = 0.05,
    jack_max = 10,
    coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
    gamma = FALSE,
    ...,
    check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities.
tree	An ultrametric, phylogenetic tree. May be an object of class phylo_divent, ape::phylo, ade4::phylog or stats::hclust.
q	The order of diversity.
...	Unused.
normalize	If TRUE, phylogenetic is normalized: the height of the tree is set to 1.
estimator	An estimator of entropy.
level	The level of interpolation or extrapolation. It may be a sample size (an integer) or a sample coverage (a number between 0 and 1). If not NULL, the asymptotic estimator is ignored.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see probabilities). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see probabilities). Used only for extrapolation.
richness_estimator	
	An estimator of richness to evaluate the total number of species, see div_richness. Used for interpolation and extrapolation.

jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by coverage.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
gamma	If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

Bias correction requires the number of individuals. See div_hill for estimators.

Entropy can be estimated at a specified level of interpolation or extrapolation, either a chosen sample size or sample coverage (Chao et al. 2014), rather than its asymptotic value. See accum_tsallis for details.

Value

A tibble with the site names, the estimators used and the estimated entropy.

References

Chao A, Gotelli NJ, Hsieh TC, Sander EL, Ma KH, Colwell RK, Ellison AM (2014). "Rarefaction and Extrapolation with Hill Numbers: A Framework for Sampling and Estimation in Species Diversity Studies." *Ecological Monographs*, **84**(1), 45–67. doi:10.1890/130133.1.

Examples

```
# Entropy of each community
ent_phylo(paracou_6_abd, tree = paracou_6_taxo, q = 2)
# Gamma entropy
ent_phylo(paracou_6_abd, tree = paracou_6_taxo, q = 2, gamma = TRUE)

# At 80% coverage
ent_phylo(paracou_6_abd, tree = paracou_6_taxo, q = 2, level = 0.8)
```

---

ent_rao   *Rao's Quadratic Entropy of a Community*

Description

Estimate the quadratic entropy (Rao 1982) of species from abundance or probability data. An estimator (Lande 1996) is available to deal with incomplete sampling.

Usage

```
ent_rao(x, ...)

## S3 method for class 'numeric'
ent_rao(
  x,
  distances = NULL,
  tree = NULL,
  normalize = TRUE,
  estimator = c("Lande", "naive"),
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
ent_rao(
  x,
  distances = NULL,
  tree = NULL,
  normalize = TRUE,
  estimator = c("Lande", "naive"),
  gamma = FALSE,
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities.
...	Unused.
distances	A distance matrix or an object of class stats::dist
tree	An ultrametric, phylogenetic tree. May be an object of class phylo_divent, ape::phylo, ade4::phylog or stats::hclust.
normalize	If TRUE, phylogenetic is normalized: the height of the tree is set to 1.
estimator	An estimator of entropy.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
gamma	If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

Rao's entropy is phylogenetic or similarity-based entropy of order 2. ent_phylo and ent_similarity with argument q = 2 provide more estimators and allow estimating entropy at a chosen level.

All species of the `species_distribution` must be found in the matrix of `distances` if it is named. If it is not or if `x` is numeric, its size must equal the number of species. Then, the order of species is assumed to be the same as that of the `species_distribution` or its numeric equivalent.

Value

A tibble with the site names, the estimators used and the estimated entropy.

References

Lande R (1996). "Statistics and Partitioning of Species Diversity, and Similarity among Multiple Communities." *Oikos*, **76**(1), 5–13. doi:10.2307/3545743.

Rao CR (1982). "Diversity and Dissimilarity Coefficients: A Unified Approach." *Theoretical Population Biology*, **21**, 24–43. doi:10.1016/00405809(82)900041.

Examples

```
# Entropy of each community
ent_rao(paracou_6_abd, tree = paracou_6_taxo)
# Similar to (but estimators are not the same)
ent_phylo(paracou_6_abd, tree = paracou_6_taxo, q = 2)

# Functional entropy
ent_rao(paracou_6_abd, distances = paracou_6_fundist)

# gamma entropy
ent_rao(paracou_6_abd, tree = paracou_6_taxo, gamma = TRUE)
```

---

ent_shannon   *Shannon's Entropy of a Community*

Description

Estimate the entropy (Shannon 1948) of species from abundance or probability data. Several estimators are available to deal with incomplete sampling.

Usage

```
ent_shannon(x, ...)

## S3 method for class 'numeric'
ent_shannon(
  x,
 estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Marcon",
   "UnveilC", "UnveiliC", "ZhangGrabchak", "naive", "Bonachela", "Grassberger2003",
    "Holste", "Miller", "Schurmann", "ZhangHz"),
```

```
    level = NULL,
    probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
    unveiling = c("geometric", "uniform", "none"),
    richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
    jack_alpha = 0.05,
    jack_max = 10,
    coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
    as_numeric = FALSE,
    ...,
    check_arguments = TRUE
)

## S3 method for class 'species_distribution'
ent_shannon(
  x,
  estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Marcon",
   "UnveilC", "UnveiliC", "ZhangGrabchak", "naive", "Bonachela", "Grassberger2003",
     "Holste", "Miller", "Schurmann", "ZhangHz"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  gamma = FALSE,
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities.
...	Unused.
estimator	An estimator of entropy.
level	The level of interpolation or extrapolation. It may be a sample size (an integer) or a sample coverage (a number between 0 and 1). If not NULL, the asymptotic estimator is ignored.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see probabilities). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see probabilities). Used only for extrapolation.
richness_estimator	
	An estimator of richness to evaluate the total number of species, see div_richness. Used for interpolation and extrapolation.

jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by coverage.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
gamma	If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

Bias correction requires the number of individuals.

See div_hill for non-specific estimators. Shannon-specific estimators are from Miller (1955), Grassberger (2003), Schürmann (2004) and Zhang (2012). More estimators can be found in the **entropy** package.

Entropy can be estimated at a specified level of interpolation or extrapolation, either a chosen sample size or sample coverage (Chao et al. 2014), rather than its asymptotic value. See accum_tsallis for details.

Value

A tibble with the site names, the estimators used and the estimated entropy.

References

Chao A, Gotelli NJ, Hsieh TC, Sander EL, Ma KH, Colwell RK, Ellison AM (2014). "Rarefaction and Extrapolation with Hill Numbers: A Framework for Sampling and Estimation in Species Diversity Studies." *Ecological Monographs*, **84**(1), 45–67. doi:10.1890/130133.1.

Grassberger P (2003). "Entropy Estimates from Insufficient Samplings." *arXiv Physics e-prints*, **0307138**(v2).

Miller GA (1955). "Note on the Bias of Information Estimates." In Quastler H (ed.), *Information Theory in Psychology: Problems and Methods*, 95–100. Free Press, Glencoe, Ill.

Schürmann T (2004). "Bias Analysis in Entropy Estimation." *Journal of Physics A: Mathematical and General*, **37**(27), L295–L301. doi:10.1088/03054470/37/27/L02.

Shannon CE (1948). "A Mathematical Theory of Communication." *The Bell System Technical Journal*, **27**(3), 379–423, 623–656. doi:10.1002/j.15387305.1948.tb01338.x.

Zhang Z (2012). "Entropy Estimation in Turing's Perspective." *Neural Computation*, **24**(5), 1368–1389. doi:10.1162/NECO_a_00266.

Examples

```
# Entropy of each community
ent_shannon(paracou_6_abd)
# gamma entropy
ent_shannon(paracou_6_abd, gamma = TRUE)

# At 80% coverage
ent_shannon(paracou_6_abd, level = 0.8)
```

---

ent_similarity   *Similarity-Based Entropy of a Community*

Description

Estimate the entropy of species from abundance or probability data and a similarity matrix between species. Several estimators are available to deal with incomplete sampling. Bias correction requires the number of individuals.

Usage

```
ent_similarity(x, similarities, q = 1, ...)

## S3 method for class 'numeric'
ent_similarity(
  x,
  similarities = diag(length(x)),
  q = 1,
 estimator = c("UnveilJ", "Max", "ChaoShen", "MarconZhang", "UnveilC", "UnveiliC",
    "naive"),
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  sample_coverage = NULL,
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
ent_similarity(
  x,
  similarities = diag(sum(!colnames(x) %in% non_species_columns)),
  q = 1,
 estimator = c("UnveilJ", "Max", "ChaoShen", "MarconZhang", "UnveilC", "UnveiliC",
```

```
    "naive"),
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  gamma = FALSE,
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class [abundances](#) or [probabilities](#). If it is a numeric vector, then its length must equal the dimensions of the `similarities` matrix: species are assumed to be in the same order.
similarities	A similarity matrix, that can be obtained by [fun_similarity](#). Its default value is the identity matrix.
q	The order of diversity.
...	Unused.
estimator	An estimator of entropy.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see [probabilities](#)). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see [probabilities](#)). Used only for extrapolation.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by [coverage](#).
sample_coverage	
	The sample coverage of `x` calculated elsewhere. Used to calculate the gamma diversity of meta-communities, see details.
as_numeric	If `TRUE`, a number or a numeric vector is returned rather than a tibble.
check_arguments	
	If `TRUE`, the function arguments are verified. Should be set to `FALSE` to save time when the arguments have been checked elsewhere.
gamma	If `TRUE`, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

All species of the `species_distribution` must be found in the matrix of `similarities` if it is named. If it is not or if `x` is numeric, its size must equal the number of species. Then, the order of species is assumed to be the same as that of the `species_distribution` or its numeric equivalent.

Similarity-Based entropy can't be interpolated of extrapolated as of the state of the art.

Value

A tibble with the site names, the estimators used and the estimated entropy.

References

There are no references for Rd macro \insertAllCites on this help page.

Examples

```
# Similarity matrix
Z <- fun_similarity(paracou_6_fundist)
# Diversity of each community
ent_similarity(paracou_6_abd, similarities = Z, q = 2)
# gamma diversity
ent_similarity(paracou_6_abd, similarities = Z, q = 2, gamma = TRUE)
```

---

ent_simpson   *Simpson's Entropy of a Community*

Description

Estimate the entropy (Simpson 1949) of species from abundance or probability data. Several estimators are available to deal with incomplete sampling.

Usage

```
ent_simpson(x, ...)

## S3 method for class 'numeric'
ent_simpson(
  x,
 estimator = c("Lande", "UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger",
   "Marcon", "UnveilC", "UnveiliC", "ZhangGrabchak", "naive", "Bonachela", "Holste"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
```

```
ent_simpson(
  x,
  estimator = c("Lande", "UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger",
    "Marcon", "UnveilC", "UnveiliC", "ZhangGrabchak", "naive", "Bonachela", "Holste"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  gamma = FALSE,
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class [abundances](#) or [probabilities](#).
...	Unused.
estimator	An estimator of entropy.
level	The level of interpolation or extrapolation. It may be a sample size (an integer) or a sample coverage (a number between 0 and 1). If not NULL, the asymptotic estimator is ignored.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see [probabilities](#)). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see [probabilities](#)). Used only for extrapolation.
richness_estimator	
	An estimator of richness to evaluate the total number of species, see [div_richness](#). Used for interpolation and extrapolation.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by [coverage](#).
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
gamma	If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

Bias correction requires the number of individuals. See div_hill for non-specific estimators.

Simpson-specific estimator is from Lande (1996).

Entropy can be estimated at a specified level of interpolation or extrapolation, either a chosen sample size or sample coverage (Chao et al. 2014), rather than its asymptotic value. See accum_tsallis for details.

Value

A tibble with the site names, the estimators used and the estimated entropy.

References

Chao A, Gotelli NJ, Hsieh TC, Sander EL, Ma KH, Colwell RK, Ellison AM (2014). "Rarefaction and Extrapolation with Hill Numbers: A Framework for Sampling and Estimation in Species Diversity Studies." *Ecological Monographs*, **84**(1), 45–67. doi:10.1890/130133.1.

Lande R (1996). "Statistics and Partitioning of Species Diversity, and Similarity among Multiple Communities." *Oikos*, **76**(1), 5–13. doi:10.2307/3545743.

Simpson EH (1949). "Measurement of Diversity." *Nature*, **163**(4148), 688. doi:10.1038/163688a0.

Examples

```
# Entropy of each community
ent_simpson(paracou_6_abd)
# gamma entropy
ent_simpson(paracou_6_abd, gamma = TRUE)

# At 80% coverage
ent_simpson(paracou_6_abd, level = 0.8)
```

---

ent_tsallis   *Tsallis Entropy of a Community*

Description

Estimate the entropy of species from abundance or probability data. Several estimators are available to deal with incomplete sampling.

Usage

```
ent_tsallis(x, q = 1, ...)

## S3 method for class 'numeric'
ent_tsallis(
```

```
  x,
  q = 1,
 estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Marcon",
    "UnveilC", "UnveiliC", "ZhangGrabchak", "naive", "Bonachela", "Holste"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "rarefy", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  sample_coverage = NULL,
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
ent_tsallis(
  x,
  q = 1,
 estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Marcon",
    "UnveilC", "UnveiliC", "ZhangGrabchak", "naive", "Bonachela", "Holste"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "rarefy", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  gamma = FALSE,
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities.
q	The order of diversity.
...	Unused.
estimator	An estimator of entropy.
level	The level of interpolation or extrapolation. It may be a sample size (an integer) or a sample coverage (a number between 0 and 1). If not NULL, the asymptotic estimator is ignored.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see probabilities). Used only for extrapolation.

unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see probabilities). Used only for extrapolation.
richness_estimator	
	An estimator of richness to evaluate the total number of species, see div_richness. Used for interpolation and extrapolation.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by coverage.
sample_coverage	
	The sample coverage of x calculated elsewhere. Used to calculate the gamma diversity of meta-communities, see details.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
gamma	If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

Bias correction requires the number of individuals. See div_hill for estimators.

Entropy can be estimated at a specified level of interpolation or extrapolation, either a chosen sample size or sample coverage (Chao et al. 2014), rather than its asymptotic value. See accum_tsallis for details.

Value

A tibble with the site names, the estimators used and the estimated entropy.

References

Chao A, Gotelli NJ, Hsieh TC, Sander EL, Ma KH, Colwell RK, Ellison AM (2014). "Rarefaction and Extrapolation with Hill Numbers: A Framework for Sampling and Estimation in Species Diversity Studies." *Ecological Monographs*, **84**(1), 45–67. doi:10.1890/130133.1.

Examples

```
# Entropy of each community
ent_tsallis(paracou_6_abd, q = 2)
# gamma entropy
ent_tsallis(paracou_6_abd, q = 2, gamma = TRUE)

# At 80% coverage
ent_tsallis(paracou_6_abd, level = 0.8)
```

---

exp_q   *Deformed exponential*

Description

Calculate the deformed exponential of order q.

Usage

```
exp_q(x, q)
```

Arguments

x	A numeric vector or array.
q	A number.

Details

The deformed exponential is the reciprocal of the deformed logarithm (Tsallis 1994), see ln_q. It is defined as $(x(1-q)+1)^{\frac{1}{(1-q)}}$. For $q > 1$, $\ln_q(+\infty) = \frac{1}{(q-1)}$ so $\exp_q(x)$ is not defined for $x > \frac{1}{(q-1)}$.

Value

A vector of the same length as x containing the transformed values.

References

Tsallis C (1994). "What Are the Numbers That Experiments Provide?" *Química Nova*, **17**(6), 468–471.

Examples

```
curve(exp_q(x, q = 0), from = -5, to = 0, lty = 2)
curve(exp(x), from = -5, to = 0, lty= 1, add = TRUE)
curve(exp_q(x, q = 2), from = -5, to = 0, lty = 3, add = TRUE)
legend("bottomright",
  legend = c(
    expression(exp[0](x)),
    expression(exp(x)),
    expression(exp[2](x))
  ),
  lty = c(2, 1, 3),
  inset = 0.02
)
```

e_n_q   *Grassberger's expectation of n^q*

Description

Expected value of n^q when n follows a Poisson distribution of parameter n.

Usage

```
e_n_q(n, q)
```

Arguments

n	A positive integer vector.
q	A positive number.

Details

The expectation of n^q when n follows a Poisson distribution was derived by Grassberger (1988).

It is computed using the beta function. Its value is 0 for $n - q + 1 < 0$, and close to 0 when $n = q$, which is not a correct estimate: it should not be used when $q > n$.

Value

A vector of the same length as n containing the transformed values.

References

Grassberger P (1988). "Finite Sample Corrections to Entropy and Dimension Estimates." *Physics Letters A*, **128**(6-7), 369–373. doi:10.1016/03759601(88)901934.

Examples

```
n <- 10
q <- 2
# Compare
e_n_q(n, q)
# with (empirical estimation)
mean(rpois(1000, lambda = n)^q)
# and (naive estimation)
n^q
```

---

fit_rac *Fit a distribution*

Description

Fit a well-known distribution to a species distribution.

Usage

```
fit_rac(x, ...)

## S3 method for class 'numeric'
fit_rac(
  x,
  distribution = c("lnorm", "lseries", "geom", "bstick"),
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
fit_rac(
  x,
  distribution = c("lnorm", "lseries", "geom", "bstick"),
  ...,
  check_arguments = TRUE
)
```

Arguments

x   An object

...   Unused.

distribution   The distribution of species abundances. May be "lnorm" (log-normal), "lseries"
                (log-series), "geom" (geometric) or "bstick" (broken stick).

check_arguments

                If TRUE, the function arguments are verified. Should be set to FALSE to save time
                when the arguments have been checked elsewhere.

Details

[abundances](#) can be used to fit rank-abundance curves (RAC) of classical distributions:

- "lnorm" for log-normal (Preston 1948).
- "lseries" for log-series (Fisher et al. 1943).
- "geom" for geometric (Motomura 1932).
- "bstick" for broken stick (MacArthur 1957). It has no parameter, so the maximum abundance
  is returned.

Value

A tibble with the sites and the estimated distribution parameters.

References

Fisher RA, Corbet AS, Williams CB (1943). "The Relation between the Number of Species and the Number of Individuals in a Random Sample of an Animal Population." *Journal of Animal Ecology*, **12**, 42–58. doi:10.2307/1411.

MacArthur RH (1957). "On the Relative Abundance of Bird Species." *Proceedings of the National Academy of Sciences of the United States of America*, **43**(3), 293–295. doi:10.1073/pnas.43.3.293, 89566.

Motomura I (1932). "On the statistical treatment of communities." *Zoological Magazine*, **44**, 379–383.

Preston FW (1948). "The Commonness, and Rarity, of Species." *Ecology*, **29**(3), 254–283. doi:10.2307/1930989.

Examples

```
fit_rac(paracou_6_abd, distribution = "lnorm")
```

---

fun_ordinariness   *Functional ordinariness of a community*

Description

The ordinariness of a species is the average similarity of its individuals with others (Leinster and Cobbold 2012).

Usage

```
fun_ordinariness(
  species_distribution,
 similarities = diag(sum(!colnames(species_distribution) %in% non_species_columns)),
  as_numeric = FALSE,
  check_arguments = TRUE
)
```

Arguments

species_distribution

An object of class species_distribution.

similarities   A similarity matrix, that can be obtained by fun_similarity. Its default value is the identity matrix.

as_numeric   If TRUE, a number or a numeric vector is returned rather than a tibble.

check_arguments

                If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

Details

All species of the `species_distribution` must be found in the matrix of `similarities` if it is named. If it is not, its size must equal the number of species. Then, the order of species is assumed to be the same as that of the `species_distribution`.

Value

A tibble with the ordinariness of each species, or a matrix if argument `as_numeric` is TRUE.

References

Leinster T, Cobbold C (2012). "Measuring Diversity: The Importance of Species Similarity." *Ecology*, **93**(3), 477–489. doi:10.1890/102402.1.

Examples

```
fun_ordinariness(paracou_6_abd, fun_similarity(paracou_6_fundist))

# Compare with probabilities
probabilities(paracou_6_abd)
# Decrease similarities so that ordinariness is close to probability
fun_ordinariness(paracou_6_abd, fun_similarity(paracou_6_fundist, rate = 100))
```

---

   fun_similarity   *Functional similarity*

Description

Transform a distance matrix into a similarity matrix (Leinster and Cobbold 2012). Similarity between two species is defined either by a negative exponential function of their distance or by the complement to 1 of their normalized distance (such that the most distant species are 1 apart).

Usage

```
fun_similarity(distances, exponential = TRUE, rate = 1, check_arguments = TRUE)
```

Arguments

distances	A distance matrix or an object of class stats::dist
exponential	If TRUE, similarity is $e^{-r\delta}$, where r is argument rate. If FALSE, it is $1 - \delta/\max(\delta)$.
rate	The decay rate of the exponential similarity.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

Value

A similarity matrix.

References

Leinster T, Cobbold C (2012). "Measuring Diversity: The Importance of Species Similarity." *Ecology*, **93**(3), 477–489. doi:10.1890/102402.1.

Examples

```
# Similarity between Paracou 6 species
hist(fun_similarity(paracou_6_fundist))
```

---

ln_q   *Deformed logarithm*

Description

Calculate the deformed logarithm of order q.

Usage

```
ln_q(x, q)
```

Arguments

x	A numeric vector or array.
q	A number.

Details

The deformed logarithm (Tsallis 1994) is defined as $\ln_q x = \frac{(x^{(1-q)} - 1)}{(1-q)}$.

The shape of the deformed logarithm is similar to that of the regular one. $\ln_1 x = \log x$.

For $q > 1$, $\ln_q(+\infty) = \frac{1}{(q-1)}$.

Value

A vector of the same length as x containing the transformed values.

References

Tsallis C (1994). "What Are the Numbers That Experiments Provide?" *Química Nova*, **17**(6), 468–471.

Examples

```
curve(ln_q( 1/ x, q = 0), 0, 1, lty = 2, ylab = "Logarithm", ylim = c(0, 10))
curve(log(1 / x), 0, 1, lty = 1, n =1E4, add = TRUE)
curve(ln_q(1 / x, q = 2), 0, 1, lty = 3, add = TRUE)
legend("topright",
  legend = c(
    expression(ln[0](1/x)),
    expression(log(1/x)),
    expression(ln[2](1/x))
  ),
  lty = c(2, 1, 3),
  inset = 0.02
 )
```

---

metacommunity	*Aggregate communities into a metacommunity*

Description

Abundances of communities are summed according to their weights to obtain the abundances of the metacommunity.

Usage

```
metacommunity(x, name = "metacommunity", ...)

## S3 method for class 'matrix'
metacommunity(
  x,
  name = "metacommunity",
  weights = rep(1, nrow(x)),
  as_numeric = TRUE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'abundances'
metacommunity(
```

```
    x,
    name = "metacommunity",
    as_numeric = FALSE,
    ...,
    check_arguments = TRUE
)
```

Arguments

x	An object of class [abundances](#) that contains several communities or a matrix of abundances with communities in rows and species in columns.
name	The name of the metacommunity
...	Unused.
weights	The weights of the sites of the species distributions.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

Details

The total abundance of the metacommunity is by design equal to the sum of community abundances so that the information used by diversity estimators. A consequence is that equal weights lead to a metacommunity whose species abundances are the sum of community species abundances.

If community weights are not equal then the metacommunity abundances are in general not integer. Most diversity estimators can't be applied to non-integer abundances but the knowledge of the sample coverage of each community allow "ChaoShen" and "Grassberger" estimators.

Value

An object of class [abundances](#) with a single row or a named vector if as_numeric = TRUE.

Examples

```
metacommunity(paracou_6_abd)

metacommunity(paracou_6_abd)
```

---

mock_3sp   *Mock data*

Description

A simple dataset to test diversity functions. It contains 3 species with their abundances, their distance matrix and their phylogenetic tree.

Usage

```
mock_3sp_abd

mock_3sp_dist

mock_3sp_tree
```

Format

`mock_3sp_abd` is a vector, `mock_3sp_dist` a matrix and `mock_3sp_tree` an object of class ape::phylo.

An object of class `dist` of length 3.

An object of class `phylo` of length 4.

Examples

```
mock_3sp_abd
mock_3sp_dist
plot(mock_3sp_tree)
axis(2)
```

---

paracou_6 *Paracou plot 6*

Description

A community assembly. It contains number of trees per species of the plot #6 of Paracou. The plot covers 6.25 ha of tropical rainforest, divided into 4 equally-sized subplots.

Usage

```
paracou_6_abd

paracou_6_wmppp
```

Format

`paracou_6_abd` is an object of class abundances, which is also a tibble::tibble. Each line of the tibble is a subplot. `paracou_6_wmppp` is a dbmss::wmppp object, i.e. a weighted, marked planar point pattern.

An object of class wmppp (inherits from ppp) of length 6.

Details

In `paracou_6_abd` (a tibble), the "site" column contains the subplot number, "weight" contains its area and all others columns contain a species. Data are the number of trees above 10 cm diameter at breast height (DBH).

In `paracou_6_wmppp` (a point pattern), the point type is tree species and the point weight is their basal area, in square centimeters.

This dataset is from Paracou field station, French Guiana, managed by Cirad.

Source

Permanent data census of Paracou: `https://paracou.cirad.fr/`

See Also

paracou_6_taxo, paracou_6_fundist

---

paracou_6_fundist   *Functional distances between Paracou plot 6 species*

Description

A functional distance matrix of species of the dataset paracou_6_abd. Distances were computed from a trait dataset including specific leaf area, wood density, seed mass and 95th percentile of height. Gower's metric (Gower 1971) was used to obtain a distance matrix.

Usage

```
paracou_6_fundist
```

Format

A matrix.

Details

This dataset is from Paracou field station, French Guiana, managed by Cirad.

Source

Permanent data census of Paracou: `https://paracou.cirad.fr/`

See Also

paracou_6_abd, paracou_6_taxo

---

paracou_6_taxo   *Taxonomy of Paracou plot 6 species*

Description

The taxonomy of species of the dataset paracou_6_abd. Distances in the tree are 1 (different species of the same genus), 2 (same family) or 3 (different families).

Usage

```
paracou_6_taxo
```

Format

An object of class ape::phylo, which is a phylogenetic tree.

Details

This dataset is from Paracou field station, French Guiana, managed by Cirad.

Source

Permanent data census of Paracou: `https://paracou.cirad.fr/`

See Also

paracou_6_abd, paracou_6_fundist

---

phylo_divent   *Class phylo_divent*

Description

Methods for dendrograms of class "phylo_divent".

Usage

```
as_phylo_divent(tree)

is_phylo_divent(x)
```

Arguments

tree	An ultrametric, phylogenetic tree. May be an object of class phylo_divent, ape::phylo, ade4::phylog or stats::hclust.
x	An object of class "phylo_divent".

Details

as_phylo_divent calculates cuts and intervals of a phylogenetic tree and makes it available both in stats::hclust and ape::phylo formats. The conversion preprocesses the tree: it calculates cuts so that the tree can be reused efficiently by phylodiversity functions.

Value

as_phylo_divent returns a phylogenetic tree that is an object of class "phylo_divent".

Examples

```
# Paracou plot 6 species taxonomy
tree <- as_phylo_divent(mock_3sp_tree)
plot(tree)
```

---

plot.phylo_divent   *Plot phylo_divent Objects*

Description

Plot objects of class "phylo_divent" produced by as_phylo_divent, that are phylogenetic trees.

Usage

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

Arguments

x	An object of class "phylo_divent".
...	Arguments passed to stats::plot.dendrogram.

Value

NULL. Called for side effects.

Examples

```
# Paracou plot 6 species taxonomy
tree <- as_phylo_divent(paracou_6_taxo)
plot(tree, leaflab = "none")
```

plot.species_distribution
Plot Profile Objects

Description

Plot objects of class "species_distribution" produced by species_distribution and similar functions.

Usage

```
## S3 method for class 'species_distribution'
plot(
  x,
  type = c("RAC", "Metacommunity"),
  ...,
  fit_rac = FALSE,
  distribution = c("lnorm", "lseries", "geom", "bstick"),
  ylog = "y",
  main = NULL,
  xlab = "Rank",
  ylab = NULL,
  palette = "Set1"
)

## S3 method for class 'species_distribution'
autoplot(
  object,
  ...,
  fit_rac = FALSE,
  distribution = c("lnorm", "lseries", "geom", "bstick"),
  ylog = TRUE,
  main = NULL,
  xlab = "Rank",
  ylab = NULL,
  pch = ggplot2::GeomPoint$default_aes$shape,
  cex = ggplot2::GeomPoint$default_aes$size
)
```

Arguments

x	An object.
type	The type of plot. "RAC" (Rank-abundance curve, or Whittaker plot) or "Metacommunity" to represent species abundances of each community along with those of the metacommunity.
...	Additional arguments to be passed to plot. Unused elsewhere.
fit_rac	If TRUE, estimate a theoretical distribution and fit the data with it. RAC plot only.

distribution	The distribution of species abundances. May be "lnorm" (log-normal), "lseries" (log-series), "geom" (geometric) or "bstick" (broken stick). RAC plot only.
ylog	If TRUE, the Y-axis is in log-scale. RAC plot only.
main	The title of the plot.
xlab	The label of the X-axis. RAC plot only.
ylab	The label of the Y-axis.
palette	The name of a color palette, recognized by [RColorBrewer::brewer.pal](#). RAC plot only.
object	An object of class [species_distribution](#).
pch	The plotting characters. See [graphics::points](#).
cex	The character expansion (size) of the points. See [graphics::points](#).

Value

NULL. Called for side effects.

---

probabilities   *Probabilities of Species*

Description

Estimate actual probabilities of species from a sample

Usage

```
probabilities(x, ...)

## S3 method for class 'numeric'
probabilities(
  x,
  estimator = c("naive", "Chao2013", "Chao2015", "ChaoShen"),
  unveiling = c("none", "uniform", "geometric"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "rarefy", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  q = 0,
  as_numeric = FALSE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'abundances'
probabilities(
  x,
```

```
    estimator = c("naive", "Chao2013", "Chao2015", "ChaoShen"),
    unveiling = c("none", "uniform", "geometric"),
    richness_estimator = c("jackknife", "iChao1", "Chao1", "rarefy", "naive"),
    jack_alpha = 0.05,
    jack_max = 10,
    coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
    q = 0,
    ...,
    check_arguments = TRUE
)
```

Arguments

x	An object. It may be:

- a numeric vector containing abundances. It may be named to track species names.
- an object of class species_distribution.

...	Unused.
estimator	One of the estimators of a probability distribution: "naive" (the default value), or "Chao2013", "Chao2015", "ChaoShen" to estimate the probabilities of the observed species in the asymptotic distribution.
unveiling	One of the possible unveiling methods to estimate the probabilities of the unobserved species: "none" (default, no species is added), "uniform" (all unobserved species have the same probability) or "geometric" (the unobserved species distribution is geometric).
richness_estimator	
	An estimator of richness to evaluate the total number of species. "jackknife" is the default value. An alternative is "rarefy" to estimate the number of species such that the entropy of the asymptotic distribution rarefied to the observed sample size equals the actual entropy of the data.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by coverage.
q	The order of diversity. Default is 0 for richness. Used only to estimate asymptotic probability distributions when argument richness_estimator is "rarefy". Then, the number of unobserved species is fitted so that the entropy of order q of the asymptotic probability distribution at the observed sample size equals the actual entropy of the data.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

Details

probabilities() estimates a probability distribution from a sample. If the estimator is not "naive", the observed abundance distribution is used to estimate the actual probability distribution. The list of species is changed: zero-abundance species are cleared, and some unobserved species are added. First, observed species probabilities are estimated following Chao and Shen (2003), i.e. input probabilities are multiplied by the sample coverage, or according to more sophisticated models: Chao et al. (2013), a single-parameter model, or Chao and Jost (2015), a two-parameter model. The total probability of observed species equals the sample coverage. Then, the distribution of unobserved species can be unveiled: their number is estimated according to the richness_estimator. The coverage deficit (1 minus the sample coverage) is shared by the unobserved species equally (unveiling = "uniform", (Chao et al. 2013)) or according to a geometric distribution (unveiling = "geometric", (Chao and Jost 2015)).

Value

An object of class "probabilities", which is a species_distribution or a numeric vector with argument as_numeric = TRUE.

References

Chao A, Jost L (2015). "Estimating Diversity and Entropy Profiles via Discovery Rates of New Species." *Methods in Ecology and Evolution*, **6**(8), 873–882. doi:10.1111/2041210X.12349.

Chao A, Shen T (2003). "Nonparametric Estimation of Shannon's Index of Diversity When There Are Unseen Species in Sample." *Environmental and Ecological Statistics*, **10**(4), 429–443. doi:10.1023/A:1026096204727.

Chao A, Wang Y, Jost L (2013). "Entropy and the Species Accumulation Curve: A Novel Entropy Estimator via Discovery Rates of New Species." *Methods in Ecology and Evolution*, **4**(11), 1091–1100. doi:10.1111/2041210x.12108.

Examples

```
# Just transform abundances into probabilities
probabilities(paracou_6_abd)
# Estimate the distribution of probabilities from observed abundances (unveiled probabilities)
prob_unv <- probabilities(
  paracou_6_abd,
  estimator = "Chao2015",
  unveiling = "geometric",
  richness_estimator = "jackknife"
)
# Estimate entropy from the unveiled probabilities
ent_shannon(prob_unv)
# Identical to
ent_shannon(paracou_6_abd, estimator = "UnveilJ")
```

---

profile_hill   *Diversity Profile of a Community*

Description

Calculate the diversity profile of a community, i.e. diversity (Hill numbers) against its order.

Usage

```
profile_hill(x, orders = seq(from = 0, to = 2, by = 0.1), ...)

## S3 method for class 'numeric'
profile_hill(
  x,
  orders = seq(from = 0, to = 2, by = 0.1),
 estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Holste",
    "Marcon", "UnveilC", "UnveiliC", "ZhangGrabchak", "naive"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  sample_coverage = NULL,
  as_numeric = FALSE,
  n_simulations = 0,
  alpha = 0.05,
  bootstrap = c("Chao2015", "Marcon2012", "Chao2013"),
  show_progress = TRUE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
profile_hill(
  x,
  orders = seq(from = 0, to = 2, by = 0.1),
 estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Holste",
    "Marcon", "UnveilC", "UnveiliC", "ZhangGrabchak", "naive"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
```

```
  gamma = FALSE,
  n_simulations = 0,
  alpha = 0.05,
  bootstrap = c("Chao2015", "Marcon2012", "Chao2013"),
  show_progress = TRUE,
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities.
orders	The orders of diversity used to build the profile.
...	Unused.
estimator	An estimator of entropy.
level	The level of interpolation or extrapolation. It may be a sample size (an integer) or a sample coverage (a number between 0 and 1). If not NULL, the asymptotic estimator is ignored.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see probabilities). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see probabilities). Used only for extrapolation.
richness_estimator	
	An estimator of richness to evaluate the total number of species, see div_richness. Used for interpolation and extrapolation.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by coverage.
sample_coverage	
	The sample coverage of x calculated elsewhere. Used to calculate the gamma diversity of meta-communities, see details.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
n_simulations	The number of simulations used to estimate the confidence envelope of the profile.
alpha	The risk level, 5% by default, of the confidence envelope of the profile.
bootstrap	The method used to obtain the probabilities to generate bootstrapped communities from observed abundances. If "Marcon2012", the probabilities are simply the abundances divided by the total number of individuals (Marcon et al. 2012). If "Chao2013" or "Chao2015" (by default), a more sophisticated approach is used (see as_probabilities) following Chao et al. (2013) or Chao and Jost (2015).
show_progress	If TRUE, a progress bar is shown during long computations.

check_arguments

> If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

gamma

> If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

A bootstrap confidence interval can be produced by simulating communities (their number is n_simulations) with rcommunity and calculating their profiles. Simulating communities implies a downward bias in the estimation: rare species of the actual community may have abundance zero in simulated communities. Simulated diversity values are recentered so that their mean is that of the actual community.

Value

A tibble with the site names, the estimators used and the estimated diversity at each order. This is an object of class "profile" that can be plotted.

References

Chao A, Jost L (2015). "Estimating Diversity and Entropy Profiles via Discovery Rates of New Species." *Methods in Ecology and Evolution*, **6**(8), 873–882. doi:10.1111/2041210X.12349.

Chao A, Wang Y, Jost L (2013). "Entropy and the Species Accumulation Curve: A Novel Entropy Estimator via Discovery Rates of New Species." *Methods in Ecology and Evolution*, **4**(11), 1091–1100. doi:10.1111/2041210x.12108.

Marcon E, Hérault B, Baraloto C, Lang G (2012). "The Decomposition of Shannon's Entropy and a Confidence Interval for *Beta* Diversity." *Oikos*, **121**(4), 516–522. doi:10.1111/j.16000706.2011.19267.x.

Examples

```
autoplot(profile_hill(paracou_6_abd))
```

---

profile_phylo   *Phylogenetic Diversity Profile of a Community*

Description

Calculate the diversity profile of a community, i.e. its phylogenetic diversity against its order.

Usage

```
profile_phylo(x, tree, orders = seq(from = 0, to = 2, by = 0.1), ...)

## S3 method for class 'numeric'
profile_phylo(
  x,
  tree,
  orders = seq(from = 0, to = 2, by = 0.1),
  normalize = TRUE,
 estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Holste",
    "Marcon", "UnveilC", "UnveiliC", "ZhangGrabchak", "naive"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  sample_coverage = NULL,
  as_numeric = FALSE,
  n_simulations = 0,
  alpha = 0.05,
  bootstrap = c("Chao2015", "Marcon2012", "Chao2013"),
  show_progress = TRUE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
profile_phylo(
  x,
  tree,
  orders = seq(from = 0, to = 2, by = 0.1),
  normalize = TRUE,
 estimator = c("UnveilJ", "ChaoJost", "ChaoShen", "GenCov", "Grassberger", "Holste",
    "Marcon", "UnveilC", "UnveiliC", "ZhangGrabchak", "naive"),
  level = NULL,
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  gamma = FALSE,
  n_simulations = 0,
  alpha = 0.05,
  bootstrap = c("Chao2015", "Marcon2012", "Chao2013"),
  show_progress = TRUE,
```

```
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class [abundances](#) or [probabilities](#).
tree	An ultrametric, phylogenetic tree. May be an object of class [phylo_divent](#), [ape::phylo](#), [ade4::phylog](#) or [stats::hclust](#).
orders	The orders of diversity used to build the profile.
...	Unused.
normalize	If `TRUE`, phylogenetic is normalized: the height of the tree is set to 1.
estimator	An estimator of entropy.
level	The level of interpolation or extrapolation. It may be a sample size (an integer) or a sample coverage (a number between 0 and 1). If not `NULL`, the asymptotic `estimator` is ignored.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see [probabilities](#)). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see [probabilities](#)). Used only for extrapolation.
richness_estimator	
	An estimator of richness to evaluate the total number of species, see [div_richness](#). Used for interpolation and extrapolation.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by [coverage](#).
sample_coverage	
	The sample coverage of `x` calculated elsewhere. Used to calculate the gamma diversity of meta-communities, see details.
as_numeric	If `TRUE`, a number or a numeric vector is returned rather than a tibble.
n_simulations	The number of simulations used to estimate the confidence envelope of the profile.
alpha	The risk level, 5% by default, of the confidence envelope of the profile.
bootstrap	The method used to obtain the probabilities to generate bootstrapped communities from observed abundances. If "Marcon2012", the probabilities are simply the abundances divided by the total number of individuals (Marcon et al. 2012). If "Chao2013" or "Chao2015" (by default), a more sophisticated approach is used (see [as_probabilities](#)) following Chao et al. (2013) or Chao and Jost (2015).
show_progress	If `TRUE`, a progress bar is shown during long computations.

check_arguments

>   If TRUE, the function arguments are verified. Should be set to FALSE to save time
>   when the arguments have been checked elsewhere.

gamma   If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

A bootstrap confidence interval can be produced by simulating communities (their number is n_simulations)
with rcommunity and calculating their profiles. Simulating communities implies a downward bias
in the estimation: rare species of the actual community may have abundance zero in simulated
communities. Simulated diversity values are recentered so that their mean is that of the actual
community.

Value

A tibble with the site names, the estimators used and the estimated diversity at each order. This is
an object of class "profile" that can be plotted.

References

Chao A, Jost L (2015). "Estimating Diversity and Entropy Profiles via Discovery Rates of New
Species." *Methods in Ecology and Evolution*, **6**(8), 873–882. doi:10.1111/2041210X.12349.

Chao A, Wang Y, Jost L (2013). "Entropy and the Species Accumulation Curve: A Novel En-
tropy Estimator via Discovery Rates of New Species." *Methods in Ecology and Evolution*, **4**(11),
1091–1100. doi:10.1111/2041210x.12108.

Marcon E, Hérault B, Baraloto C, Lang G (2012). "The Decomposition of Shannon's Entropy and a
Confidence Interval for *Beta* Diversity." *Oikos*, **121**(4), 516–522. doi:10.1111/j.16000706.2011.19267.x.

Examples

```
profile_phylo(paracou_6_abd, tree = paracou_6_taxo)
```

---

profile_similarity   *Similarity-Based Diversity Profile of a Community*

Description

Calculate the diversity profile of a community, i.e. its similarity-based diversity against its order.

Usage

```
profile_similarity(
  x,
  similarities,
  orders = seq(from = 0, to = 2, by = 0.1),
  ...
)

## S3 method for class 'numeric'
profile_similarity(
  x,
  similarities = diag(length(x)),
  orders = seq(from = 0, to = 2, by = 0.1),
 estimator = c("UnveilJ", "Max", "ChaoShen", "MarconZhang", "UnveilC", "UnveiliC",
    "naive"),
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  richness_estimator = c("jackknife", "iChao1", "Chao1", "naive"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  sample_coverage = NULL,
  as_numeric = FALSE,
  n_simulations = 0,
  alpha = 0.05,
  bootstrap = c("Chao2015", "Marcon2012", "Chao2013"),
  show_progress = TRUE,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'species_distribution'
profile_similarity(
  x,
  similarities = diag(sum(!colnames(x) %in% non_species_columns)),
  orders = seq(from = 0, to = 2, by = 0.1),
 estimator = c("UnveilJ", "Max", "ChaoShen", "MarconZhang", "UnveilC", "UnveiliC",
    "naive"),
  probability_estimator = c("Chao2015", "Chao2013", "ChaoShen", "naive"),
  unveiling = c("geometric", "uniform", "none"),
  jack_alpha = 0.05,
  jack_max = 10,
  coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
  gamma = FALSE,
  n_simulations = 0,
  alpha = 0.05,
  bootstrap = c("Chao2015", "Marcon2012", "Chao2013"),
  show_progress = TRUE,
```

```
  ...,
  check_arguments = TRUE
)
```

Arguments

x	An object, that may be a numeric vector containing abundances or probabilities, or an object of class abundances or probabilities.
similarities	A similarity matrix, that can be obtained by fun_similarity. Its default value is the identity matrix.
orders	The orders of diversity used to build the profile.
...	Unused.
estimator	An estimator of entropy.
probability_estimator	
	A string containing one of the possible estimators of the probability distribution (see probabilities). Used only for extrapolation.
unveiling	A string containing one of the possible unveiling methods to estimate the probabilities of the unobserved species (see probabilities). Used only for extrapolation.
richness_estimator	
	An estimator of richness to evaluate the total number of species, see div_richness. Used for interpolation and extrapolation.
jack_alpha	The risk level, 5% by default, used to optimize the jackknife order.
jack_max	The highest jackknife order allowed. Default is 10.
coverage_estimator	
	An estimator of sample coverage used by coverage.
sample_coverage	
	The sample coverage of x calculated elsewhere. Used to calculate the gamma diversity of meta-communities, see details.
as_numeric	If TRUE, a number or a numeric vector is returned rather than a tibble.
n_simulations	The number of simulations used to estimate the confidence envelope of the profile.
alpha	The risk level, 5% by default, of the confidence envelope of the profile.
bootstrap	The method used to obtain the probabilities to generate bootstrapped communities from observed abundances. If "Marcon2012", the probabilities are simply the abundances divided by the total number of individuals (Marcon et al. 2012). If "Chao2013" or "Chao2015" (by default), a more sophisticated approach is used (see as_probabilities) following Chao et al. (2013) or Chao and Jost (2015).
show_progress	If TRUE, a progress bar is shown during long computations.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
gamma	If TRUE, γ diversity, i.e. diversity of the metacommunity, is computed.

Details

A bootstrap confidence interval can be produced by simulating communities (their number is `n_simulations`) with [rcommunity](#) and calculating their profiles. Simulating communities implies a downward bias in the estimation: rare species of the actual community may have abundance zero in simulated communities. Simulated diversity values are recentered so that their mean is that of the actual community.

Value

A tibble with the site names, the estimators used and the estimated diversity at each order. This is an object of class "profile" that can be plotted.

References

Chao A, Jost L (2015). "Estimating Diversity and Entropy Profiles via Discovery Rates of New Species." *Methods in Ecology and Evolution*, **6**(8), 873–882. doi:10.1111/2041210X.12349.

Chao A, Wang Y, Jost L (2013). "Entropy and the Species Accumulation Curve: A Novel Entropy Estimator via Discovery Rates of New Species." *Methods in Ecology and Evolution*, **4**(11), 1091–1100. doi:10.1111/2041210x.12108.

Marcon E, Hérault B, Baraloto C, Lang G (2012). "The Decomposition of Shannon's Entropy and a Confidence Interval for *Beta* Diversity." *Oikos*, **121**(4), 516–522. doi:10.1111/j.16000706.2011.19267.x.

Examples

```
# Similarity matrix
Z <- fun_similarity(paracou_6_fundist)
# Profile
profile_similarity(paracou_6_abd, similarities = Z, q = 2)
```

---

rcommunity   *Random communities*

Description

Draw random communities according to a probability distribution.

Usage

```
rcommunity(
  n,
  size = sum(abd),
  prob = NULL,
  abd = NULL,
  bootstrap = c("Chao2015", "Marcon2012", "Chao2013"),
```

```
    species_number = 300,
    distribution = c("lnorm", "lseries", "geom", "bstick"),
    sd_lnorm = 1,
    prob_geom = 0.1,
    alpha_lseries = 40,
    coverage_estimator = c("ZhangHuang", "Chao", "Turing", "Good"),
    check_arguments = TRUE
)
```

Arguments

`n`	The number of communities to draw.
`size`	The number of individuals to draw in each community.
`prob`	A numeric vector containing probabilities.
`abd`	A numeric vector containing abundances.
`bootstrap`	The method used to obtain the probabilities to generate bootstrapped communities from observed abundances. If "Marcon2012", the probabilities are simply the abundances divided by the total number of individuals (Marcon et al. 2012). If "Chao2013" or "Chao2015" (by default), a more sophisticated approach is used (see as_probabilities) followingChao et al. (2013) or Chao and Jost (2015).
`species_number`	The number of species.
`distribution`	The distribution of species abundances. May be "lnorm" (log-normal), "lseries" (log-series), "geom" (geometric) or "bstick" (broken stick).
`sd_lnorm`	The simulated log-normal distribution standard deviation. This is the standard deviation on the log scale.
`prob_geom`	The proportion of ressources taken by successive species of the geometric distribution.
`alpha_lseries`	Fisher's alpha in the log-series distribution.
`coverage_estimator`	
	An estimator of sample coverage used by coverage.
`check_arguments`	
	If `TRUE`, the function arguments are verified. Should be set to `FALSE` to save time when the arguments have been checked elsewhere.

Details

Communities of fixed `size` are drawn in a multinomial distribution according to the distribution of probabilities provided by `prob`. An abundance vector `abd` may be used instead of probabilities, then `size` is by default the total number of individuals in the vector. Random communities can be built by drawing in a multinomial law following Marcon et al. (2012), or trying to estimate the distribution of the actual community with as_probabilities. If `bootstrap` is "Chao2013", the distribution is estimated by a single parameter model and unobserved species are given equal probabilities. If `bootstrap` is "Chao2015", a two-parameter model is used and unobserved species follow a geometric distribution.

Alternatively, the probabilities may be drawn following a classical distribution: either lognormal ("lnorm") (Preston 1948) with given standard deviation (`sd_lnorm`; note that the mean is actually

a normalizing constant. Its values is set equal to 0 for the simulation of the normal distribution of unnormalized log-abundances), log-series ("lseries") (Fisher et al. 1943) with parameter `alpha`, geometric ("geom") one (Motomura 1932) with parameter `prob_geom`, or broken stick ("bstick") (MacArthur 1957). The number of simulated species is fixed by `species_number`, except for "lseries" where it is obtained from `alpha_lseries` and `size`: $S = \alpha \ln(1 + size/\alpha)$.

Log-normal, log-series and broken-stick distributions are stochastic. The geometric distribution is completely determined by its parameters.

Value

An object of class abundances.

Examples

```
# Generate a community made of 100000 individuals among 300 species and fit it
abundances <- rcommunity(n = 1, size = 1E5,
  species_number = 300, distribution = "lnorm")
autoplot(abundances)
```

---

rlseries   *Log-Series Distribution*

Description

Random generation for the log-series distribution.

Usage

```
rlseries(n, size, alpha, show_progress = TRUE, check_arguments = TRUE)
```

Arguments

n	Number of observations.
size	The size of the distribution.
alpha	Fisher's α.
show_progress	If TRUE, a progress bar is shown during long computations.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

Details

Fast implementation of the random generation of a log-series distribution (Fisher et al. 1943).

The complete set of functions (including density, distribution function and quantiles) can be found in package *sads* but this implementation of the random generation is much faster.

If `size` is too large, i.e. `size + 1` can't be distinguished from `size` due to rounding, then an error is raised.

Value

A numeric vector with the random values drawn from the log-series distribution.

References

Fisher RA, Corbet AS, Williams CB (1943). "The Relation between the Number of Species and the Number of Individuals in a Random Sample of an Animal Population." *Journal of Animal Ecology*, **12**, 42–58. doi:10.2307/1411.

Examples

```
# Generate a community made of 10000 individuals with alpha=40
size <- 1E4
alpha <- 40
species_number <- -alpha * log(alpha / (size + alpha))
abundances <- rlseries(species_number, size = 1E5, alpha = 40)
# rCommunity() may be a better choice
autoplot(rcommunity(n = 1, size = 1E4, alpha = 40, distribution = "lseries"))
```

---

species_distribution   *Species Distributions*

Description

A Species Distribution is a tibble::tibble containing species abundances or probabilities. Rows of the tibble are communities and column are species. Values are either abundances or probabilities. Special columns contain the site names, and their weights (e.g. their area or number of individuals): their names must be "site" and "weight". All other column names are considered as species names.

Usage

```
species_distribution(x, names = NULL, weights = NULL, check_arguments = TRUE)

as_species_distribution(x, ...)

## S3 method for class 'numeric'
as_species_distribution(x, ..., check_arguments = TRUE)

## S3 method for class 'matrix'
as_species_distribution(
  x,
  names = NULL,
  weights = NULL,
  ...,
  check_arguments = TRUE
)
```

```
## S3 method for class 'data.frame'
as_species_distribution(x, ..., check_arguments = TRUE)

is_species_distribution(x)

as_probabilities(x, ...)

## S3 method for class 'numeric'
as_probabilities(x, ..., check_arguments = TRUE)

## S3 method for class 'matrix'
as_probabilities(x, names = NULL, weights = NULL, ..., check_arguments = TRUE)

## S3 method for class 'data.frame'
as_probabilities(x, ..., check_arguments = TRUE)

is_probabilities(x)

abundances(
  x,
  round = TRUE,
  names = NULL,
  weights = NULL,
  check_arguments = TRUE
)

as_abundances(x, ...)

## S3 method for class 'numeric'
as_abundances(x, round = TRUE, ..., check_arguments = TRUE)

## S3 method for class 'matrix'
as_abundances(
  x,
  round = TRUE,
  names = NULL,
  weights = NULL,
  ...,
  check_arguments = TRUE
)

## S3 method for class 'data.frame'
as_abundances(x, ..., check_arguments = TRUE)

is_abundances(x)

## S3 method for class 'species_distribution'
as.matrix(x, use.names = TRUE, ...)
```

```
## S3 method for class 'species_distribution'
as.double(x, use.names = TRUE, ...)

## S3 method for class 'species_distribution'
as.numeric(x, use.names = TRUE, ...)
```

Arguments

x	an object.
names	The names of the species distributions.
weights	The weights of the sites of the species distributions.
check_arguments	
	If TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
...	Unused.
round	If TRUE, the values of x are converted to integers.
use.names	If TRUE, the names of the species_distribution are kept in the matrix or vector they are converted to.

Details

species_distribution objects include abundances and probabilities objects.

as_species_distribution(), as_abundances() and as_probabilities format the numeric, matrix or dataframe x so that appropriate versions of community functions (generic methods such as [plot](#) or [div_richness](#)) are applied. Abundance values are rounded (by default) to the nearest integer.

as_probabilities() normalizes the vector x so that it sums to 1. It gives the same output as probabilities() with estimator = "naive".

species_distribution objects objects can be plotted by [plot](#) and [autoplot](#).

Value

An object of classes "species_distribution" and "abundances" or "probabilities".

as.double() and its synonymous as.numeric() return a numeric vector that contains species abundances or probabilities of a single-row species_distribution. as.matrix() returns a numeric matrix if the species_distribution contains several rows. These are methods of the generic functions for class species_distribution.

References

There are no references for Rd macro \insertAllCites on this help page.

Examples

```
# Paracou data is a tibble
paracou_6_abd
# Class
class(paracou_6_abd)
is_species_distribution(paracou_6_abd)
# Whittaker plot fitted by a log-normal distribution
autoplot(paracou_6_abd[1,], fit_rac = TRUE, distribution = "lnorm")
```

Index