

Package: DiversityStats (via r-universe)

May 15, 2026

Title Diversity Indices with Statistical Inference

Version 0.1.0.9000

Description Provides a comprehensive framework for analyzing diversity from frequency/abundance count data. Implements a wide range of classical and entropy-based diversity indices, including Berger-Parker, Simpson (and related variants), Shannon, Brillouin, McIntosh, Margalef, Menhinick and Smith-Wilson. Supports permutation-based hypothesis tests for comparing groups with respect to diversity (global and pairwise comparisons), as well as confidence interval estimation using multiple bootstrap methods. Includes functionality for generating diversity profiles based on parametric families such as Hill numbers, Rényi entropy, and Tsallis entropy. The methods are applicable to ecological community data (species abundance counts) and genetic or phenotypic class frequency data.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.3

BuildManual TRUE

Imports boot, dplyr, mathjaxr, multcompView, Rdpack, stats, tidyr, utils

Suggests cluster, EvaluateCore, ggplot2, knitr, pander, rmarkdown, vegan

RdMacros mathjaxr, Rdpack

Copyright 2025-2026, ICAR-NBPGR

URL <https://cran.r-project.org/package=PGRdup>,
<https://github.com/aravind-j/DiversityStats>,
<https://doi.org/10.5281/zenodo.20056661>,
<https://aravind-j.github.io/DiversityStats/>

BugReports <https://github.com/aravind-j/DiversityStats/issues>

Depends R (>= 3.5)
Config/pak/sysreqs libicu-dev
Repository https://aravind-j.r-universe.dev
Date/Publication 2026-05-15 12:53:40 UTC
RemoteUrl https://github.com/aravind-j/diversitystats
RemoteRef HEAD
RemoteSha 16f2299579d11f5c301fe8f9b8f523d123d51ae6

Contents

bootstrap.ci	2
diversity.calc	5
diversity.compare	13
diversity.profile	16
diversity_functions	21
permutation_tests	23

Index **26**

bootstrap.ci	<i>Bootstrap Confidence Intervals</i>
--------------	---------------------------------------

Description

This function generates bootstrap resamples using `boot` and computes confidence intervals using several standard bootstrap methods via `boot.ci`. The indexing for the statistic function is handled internally.

Usage

```

bootstrap.ci(
  x,
  fun,
  R = 1000,
  conf = 0.95,
  na.omit = TRUE,
  type = c("norm", "basic", "stud", "perc", "bca"),
  parallel = c("no", "multicore", "snow"),
  ncpus = getOption("boot.ncpus", 1L),
  cl = NULL,
  seed = 123,
  ...
)
  
```

Arguments

<code>x</code>	A numeric or factor vector of observations.
<code>fun</code>	A function to summarize the observations.
<code>R</code>	Integer specifying the number of permutations. Default is 1000.
<code>conf</code>	Confidence level of the interval. Default is 0.95.
<code>na.omit</code>	logical. If TRUE, when <code>x</code> is a factor, missing values (NA) are ignored and not included as a distinct factor level for computation. Default is TRUE.
<code>type</code>	A vector of character strings representing the type of intervals required. The value should be any subset of the values <code>c("norm", "basic", "stud", "perc", "bca")</code> or simply <code>"all"</code> which will compute all five types of intervals.
<code>parallel</code>	The type of parallel operation to be used (if any). If missing, the default is taken from the option <code>"boot.parallel"</code> (and if that is not set, <code>"no"</code>).
<code>ncpus</code>	integer: number of processes to be used in parallel operation: typically one would chose this to the number of available CPUs.
<code>cl</code>	An optional parallel or snow cluster for use if <code>parallel = "snow"</code> . If not supplied, a cluster on the local machine is created for the duration of the boot call.
<code>seed</code>	Integer. Random seed used to ensure reproducibility of bootstrap. Default is 123.
<code>...</code>	Additional arguments passed to <code>fun</code> .

Details

Supported interval types include normal approximation, basic, studentized (bootstrap-t), percentile, and bias-corrected and accelerated (BCa) intervals. If a requested interval type cannot be computed (for example, studentized or BCa intervals), the function falls back to percentile intervals.

Value

A a named list of confidence intervals, each containing lower and upper bounds, with additional attributes storing the observed statistic and the mean of the bootstrap replicates.

Note

The default number of bootstrap replicates $R = 1000$ is provided for quick exploratory analysis. For more reliable (but slower) confidence intervals or standard error estimates it is strongly recommended to increase R to at 5000-10000, depending on your data and required precision.

Examples

```
library(EvaluateCore)

pdata <- cassava_CC

qual <- c("QUAL", "LNGS", "PTLC", "DSTA", "LFRT", "LBTEF", "CBTR", "NMLB",
         "ANGB", "QUAL9M", "LVC9M", "TNPR9M", "PL9M", "STRP", "STRC",
         "PSTR")
```

```

# Conver '#t qualitative data columns to factor
pdata[, qual] <- lapply(pdata[, qual], as.factor)

str(pdata)

# NOTE: Increase R to 10000 for more reliable (but slower) estimates.

# Bootstrap CIs ----

bootstrap.ci(pdata$NMSR, mean, type = "norm", R = 100)
bootstrap.ci(pdata$NMSR, mean, type = "basic", R = 100)
bootstrap.ci(pdata$NMSR, mean, type = "perc", R = 100)
bootstrap.ci(pdata$NMSR, mean, type = "bca", R = 100)

bootstrap.ci(pdata$NMSR, mean,
             type = c("norm", "basic", "perc", "bca"),
             R = 100)

bootstrap.ci(pdata$LNCS, shannon, type = "norm", R = 100)
bootstrap.ci(pdata$PTLC, simpson, type = "basic", R = 100)
bootstrap.ci(pdata$LFRT, mcintosh_evenness, type = "perc", R = 100)
bootstrap.ci(pdata$LBTEF, mcintosh_diversity, type = "bca", R = 100)

bootstrap.ci(pdata$LNCS, shannon,
             type = c("norm", "basic", "perc", "bca"),
             R = 100, base = 2)

# Studentised intervals require a `fun` returning
# variances in addition to an estimate

bootstrap.ci(pdata$NMSR, mean, type = "stud", R = 100)

stat_fun_mean <- function(x) {
  est <- mean(x)
  se <- sd(x) / sqrt(length(x))
  out <- c(est, se)
  # Important : Tells bootstrap.ci to consider second output as SE
  attr(out, "se") <- TRUE
  return(out)
}

bootstrap.ci(pdata$NMSR, stat_fun_mean, type = "stud", R = 100)

bootstrap.ci(pdata$DSTA, shannon, type = "stud", R = 100)

stat_fun_shannon <- function(x, base = 2) {
  tab <- tabulate(x)
  p <- tab / length(x)
  # Only keep p > 0 to avoid log(0)
  p <- p[p > 0]
  est <- -sum(p * log(p, base = base))
  # Approximate SE using sqrt(Var(p * log(p)))

```

```

se <- sqrt(sum((p * log(p, base = base))^2) / length(x))
out <- c(est, se)
# Important : Tells bootstrap.ci to consider second output as SE
attr(out, "se") <- TRUE
return(out)
}

bootstrap.ci(pdata$DSTA, stat_fun_shannon, type = "stud", R = 100)

```

diversity.calc

*Compute Diversity Measures***Description**

The function `diversity.calc` calculates the following diversity measures

- Margalef's Richness Index ($D_{Margalef}$) (Margalef 1973)
- Menhinick's Index ($D_{Menhinick}$) (Menhinick 1964)
- Berger-Parker Index (D_{BP}) (Berger and Parker 1970)
- Reciprocal Berger-Parker Index (D_{BPR}) (Magurran 2011)
- Simpson's Index (d) (Simpson 1949; Peet 1974)
- Simpson's Index of Diversity or Gini's Diversity Index or Gini-Simpson Index or Nei's Diversity Index or Nei's Variation Index (D) or Hurlbert's probability of interspecific encounter (PIE) (Gini 1912, 1912; Greenberg 1956; Berger and Parker 1970; Hurlbert 1971; Nei 1973; Peet 1974)
- Maximum Simpson's Index of Diversity or Maximum Nei's Diversity/Variation Index (D_{max}) (Hennink and Zeven 1990)
- Simpson's Reciprocal Index or Hill's N_2 (D_R) or Effective number of Species (ENS_d) (Williams 1964; Hill 1973)
- Relative Simpson's Index of Diversity or Relative Nei's Diversity/Variation Index (D') (Hennink and Zeven 1990)
- Simpson's evenness or equitability (D_e) (Pielou 1966; Hill 1973)
- Shannon or Shannon-Weaver or Shannon-Wiener Diversity Index or Shannon entropy (H) (Shannon and Weaver 1949; Peet 1974)
- Maximum Shannon-Weaver Diversity Index (H_{max}) (Hennink and Zeven 1990)
- Relative Shannon-Weaver Diversity Index or Shannon Equitability Index (H') or Pielou's Evenness (J) (Pielou 1966; Hennink and Zeven 1990)
- Effective number of species for the Shannon - Weaver Diversity Index (ENS_H) or Hill's N_1 (MacArthur 1965; Hill 1973)
- Heip's Evenness Index (E_{Heip}) (Heip 1974)
- McIntosh Diversity Index (D_{Mc}) (McIntosh 1967; Peet 1974)
- McIntosh Evenness Index (E_{Mc}) (Pielou 1975)

- Smith & Wilson's Evenness Index (E_{var}) (Smith and Wilson 1996)
- Brillouin Diversity Index ($D_{Brillouin}$) (Brillouin 2013)
- Rényi Entropy (${}^q H_{Renyi}$) (Renyi 1960)
- Tsallis or HCDT Entropy (${}^q H_{Tsallis}$) (Havrda and Charvat 1967; Daroczy 1970; Tsallis 1988)
- Hill Numbers (${}^q D$) (Hill 1973)

Usage

```
diversity.calc(x, base = exp(1), na.omit = TRUE)
```

Arguments

x	A factor vector of categories (e.g., species, traits). The frequency of each level is treated as the abundance of that category.
base	The logarithm base to be used for computation of shannon family of diversity indices. Default is exp(1).
na.omit	logical. If TRUE, missing values (NA) are ignored and not included as a distinct factor level for computation. Default is TRUE.

Value

A list of the different diversity measures computed.

richness The number of classes in x or the richness.

margalef_index Margalef's Richness Index ($D_{Margalef}$) (Margalef 1973)

menhinick_index Menhinick's Index ($D_{Menhinick}$) (Menhinick 1964)

berger_parker Berger-Parker Index (D_{BP}) (Berger and Parker 1970)

berger_parker_reciprocal Reciprocal Berger-Parker Index (D_{BPR}) (Magurran 2011)

simpson Simpson's index (d) (Simpson 1949; Peet 1974)

gini_simpson Simpson's Index of Diversity or Gini's Diversity Index or Gini-Simpson Index or Nei's Diversity Index or Nei's Variation Index (D) or Hurlbert's probability of interspecific encounter (PIE) (Gini 1912, 1912; Greenberg 1956; Berger and Parker 1970; Hurlbert 1971; Nei 1973; Peet 1974)

simpson_max Maximum Simpson's Index of Diversity or Maximum Nei's Diversity/Variation Index (D_{max}) (Hennink and Zeven 1990)

simpson_reciprocal Simpson's Reciprocal Index or Hill's N_2 (D_R) or Effective number of Species (ENS_d) (Williams 1964; Hill 1973)

simpson_relative Relative Simpson's Index of Diversity or Relative Nei's Diversity/Variation Index (D') (Hennink and Zeven 1990)

simpson_evenness Simpson's evenness or equitability (D_e) (Pielou 1966; Hill 1973)

shannon Shannon or Shannon-Weaver or Shannon-Wiener Diversity Index or Shannon entropy (H) (Shannon and Weaver 1949; Peet 1974)

shannon_max Maximum Shannon-Weaver Diversity Index (H_{max}) (Hennink and Zeven 1990)

- shannon_relative** Relative Shannon-Weaver Diversity Index or Shannon Equitability Index (H') or Pielou's Evenness (J) (Pielou 1966; Hennink and Zeven 1990)
- shannon_ens** Effective number of species for the Shannon - Weaver Diversity Index (ENS_H) or Hill's N_1 (Macarthur 1965; Hill 1973)
- heip_evenness** Heip's Evenness Index (E_{Heip}) (Heip 1974)
- mcintosh_diversity** McIntosh Diversity Index (D_{Mc}) (McIntosh 1967; Peet 1974)
- mcintosh_evenness** McIntosh Evenness Index (E_{Mc}) (Pielou 1975)
- smith_wilson** Smith & Wilson's Evenness Index (E_{var}) (Smith and Wilson 1996)
- brillouin_index** Brillouin Diversity Index (Brillouin 2013)
- renyi_entropy_0** Rényi Entropy of order 0 (Renyi 1960)
- renyi_entropy_1** Rényi Entropy of order 1
- renyi_entropy_2** Rényi Entropy of order 2
- tsallis_entropy_0** Tsallis Entropy of order 0 (Havrda and Charvat 1967; Daroczy 1970; Tsallis 1988)
- tsallis_entropy_1** Tsallis Entropy of order 1
- tsallis_entropy_2** Tsallis Entropy of order 2
- hill_number_0** Hill Number of order 0 (0D) (Hill 1973)
- hill_number_1** Hill Number of order 1 (1D)
- hill_number_2** Hill Number of order 2 (2D)

Details

The diversity indices implemented in `diversity.calc` are computed as follows.

Richness Indices:

The number of classes of a phenotypic trait (or species richness) (k) can be described by adjusting for sample size (N) in Margalef's Richness Index ($D_{Margalef}$) (Margalef 1973) and Menhinick's Index ($D_{Menhinick}$) (Menhinick 1964)

These are computed as follows.

$$D_{Margalef} = \frac{k - 1}{\ln(N)}$$

$$D_{Menhinick} = \frac{k}{\sqrt{N}}$$

Berger-Parker Index: This is the proportion of individuals belonging to the most abundant class in a trait (or species in a community) and is computed as below.

$$D_{BP} = \max(p_i)$$

Where, p_i denotes the proportion/fraction/frequency of accessions in the i th phenotypic class for a trait (or number of individuals in the i th species).

It's reciprocal estimates the relative diversity of this class.

$$D_{BPR} = \frac{1}{D_{BP}}$$

Simpson's and Related Indices: Simpson's index (d) which estimates the probability that two accessions randomly selected will belong to the same phenotypic class of a trait (or species in a community), is computed as follows (Simpson 1949; Peet 1974).

$$d = \sum_{i=1}^k p_i^2$$

Where, k is the number of phenotypic classes for the trait (or number of species in the community). The value of d can range from 0 to 1 with 0 representing maximum diversity and 1, no diversity. d is subtracted from 1 to give Simpson's index of diversity (D) (Greenberg 1956; Berger and Parker 1970; Hurlbert 1971; Peet 1974; Hennink and Zeven 1990) originally suggested by Gini (1912, 1912) and described in literature as Gini's diversity index or Gini-Simpson index. It is the same as Nei's diversity index or Nei's variation index (Nei 1973; Hennink and Zeven 1990). Greater the value of D , greater the diversity with a range from 0 to 1.

$$D = 1 - d$$

The maximum value of D , D_{max} occurs when accessions are uniformly distributed across the phenotypic classes (or individuals are uniformly distributed across species in a community) and is computed as follows (Hennink and Zeven 1990).

$$D_{max} = 1 - \frac{1}{k}$$

Reciprocal of d gives the Simpson's reciprocal index (D_R) (Williams 1964; Hennink and Zeven 1990) and can range from 1 to k . This was also described in Hill (1973) as N_2 or as Effective number of Species (or classes) (ENS_d).

$$D_R = \frac{1}{d}$$

Relative Simpson's index of diversity or Relative Nei's diversity/variation index (H') (Hennink and Zeven 1990) is defined as follows (Peet 1974).

$$D' = \frac{D}{D_{max}}$$

Simpson's evenness or equitability (D_e is described as follows (Pielou 1966; Hill 1973).

$$D_e = \frac{1}{d \cdot k}$$

Shannon-Weaver and Related Indices: An index of information H , was described by Shannon and Weaver (1949) as follows.

$$H = - \sum_{i=1}^k p_i \log_2(p_i)$$

H is described as Shannon or Shannon-Weaver or Shannon-Wiener diversity index or Shannon entropy in literature (Shannon and Weaver 1949; Peet 1974).

Alternatively, H is also computed using natural logarithm instead of logarithm to base 2.

$$H = - \sum_{i=1}^k p_i \ln(p_i)$$

The maximum value of H (H_{max}) is $\ln(k)$. This value occurs when each phenotypic class for a trait has the same proportion of accessions (or each species in a community has the same proportion of individuals) (Hennink and Zeven 1990).

$$H_{max} = \log_2(k) \quad \text{OR} \quad H_{max} = \ln(k)$$

The relative Shannon-Weaver diversity index or Shannon equitability index (H') or Pielou's Evenness (J) is the Shannon diversity index (H) divided by the maximum diversity (H_{max}) (Pielou 1966; Hennink and Zeven 1990).

$$H' = \frac{H}{H_{max}}$$

Macarthur (1965) described the Effective number of species (or classes) for the Shannon index (ENS_H) as follows.

$$ENS_H = e^H$$

Heip's index or Heip's Evenness index is a transformation of the Shannon–Wiener diversity index that standardizes it relative to number of classes in the trait (or species richness) and is computed as follows (Heip 1974).

$$E_{Heip} = \frac{e^H - 1}{k - 1}$$

McIntosh's Measure of Diversity: A similar index of diversity was described by McIntosh (1967) as follows (D_{Mc}) (Peet 1974).

$$D_{Mc} = \frac{N - \sqrt{\sum_{i=1}^k n_i^2}}{N - \sqrt{N}}$$

Where, n_i denotes the number of accessions in the i th phenotypic class for a trait (or number of individuals in the i th species in the community) and N is the total number of accessions so that $p_i = n_i/N$.

An additional measure of evenness was proposed by Pielou (1975) as follows.

$$E_{Mc} = \frac{N - \sqrt{\sum_{i=1}^k n_i^2}}{N - \frac{N}{\sqrt{S}}}$$

Smith & Wilson's Evenness Index: This index measures how equally are accessions/genotypes distributed among different trait classes (or individuals individuals are distributed among different species in a community). This is less sensitive to rare classes or species and is computed as follows.

$$E_{var} = 1 - \frac{2}{\pi} \arctan \left(\frac{1}{k} \sum_{i=1}^k (\ln n_i - \overline{\ln n})^2 \right)$$

Brillouin Diversity Index: This is an information-theoretic measure appropriate for complete censuses and is computed as follows (Brillouin 2013).

$$H_B = \frac{\ln(N!) - \sum \ln(n_i!)}{N}$$

Parametric Indices:

Parametric indices, also known as multivariate or compound indices, use a sensitivity parameter (q) to weigh frequent and rare classes within a trait (or common or rare species within a community).

The Rényi entropy extends several entropy measures, including Shannon entropy, and is computed as follows (Rényi 1960).

$${}^q H_{Renyi} = \frac{1}{1-q} \ln \sum_{i=1}^k p_i^q, \quad q \geq 0, q \neq 1$$

It is more frequently computed using natural logarithm instead of logarithm to base 2. The index is undefined for ($q = 1$), but Shannon entropy is as a limiting case.

Tsallis proposed a similar measure, the HCDT or Tsallis entropy (Havrda and Charvat 1967; Daroczy 1970; Tsallis 1988), which matches species richness for $q = 0$, Shannon entropy $q = 1$, and the Gini-Simpson index for $q = 2$.

$${}^q H_{Tsallis} = \frac{1}{q-1} \left(1 - \sum_{i=1}^k p_i^q \right), \quad q \geq 0, q \neq 1$$

Hill showed that species richness, Shannon entropy, and Simpson's index are all related diversity indices, collectively known as Hill numbers which is defined as below (Hill 1973).

$${}^q D = \left(\sum_{i=1}^k p_i^q \right)^{\frac{1}{1-q}}, \quad q \geq 0, q \neq 1$$

Where,

$${}^0 D = k$$

$${}^1 D = e^H$$

$${}^2 D = D_R$$

References

Berger WH, Parker FL (1970). "Diversity of planktonic foraminifera in deep-sea sediments." *Science*, **168**(3937), 1345–1347.

Brillouin L (2013). *Science and information theory*, Dover edition. Dover Publications, Inc., Mineola, New York. ISBN 978-0-486-31641-3.

- Daroczy Z (1970). "Generalized information functions." *Information and Control*, **16**(1), 36–51.
- Gini C (1912). *Variabilita e Mutabilita. Contributo allo Studio delle Distribuzioni e delle Relazioni Statistiche. [Fasc. I.]*. Tipogr. di P. Cuppini, Bologna.
- Gini C (1912). "Variabilita e mutabilita." In Pizetti E, Salvemini T (eds.), *Memorie di Metodologica Statistica*. Liberia Eredi Virgilio Veschi, Roma, Italy.
- Greenberg JH (1956). "The measurement of linguistic diversity." *Language*, **32**(1), 109.
- Havrda J, Charvat F (1967). "Quantification method of classification processes. Concept of structural α -entropy." *Kybernetika*, **3**(1), (30)–35.
- Heip C (1974). "A new index measuring evenness." *Journal of the Marine Biological Association of the United Kingdom*, **54**(3), 555–557.
- Hennink S, Zeven AC (1990). "The interpretation of Nei and Shannon-Weaver within population variation indices." *Euphytica*, **51**(3), 235–240.
- Hill MO (1973). "Diversity and evenness: A unifying notation and its consequences." *Ecology*, **54**(2), 427–432.
- Hurlbert SH (1971). "The nonconcept of species diversity: a critique and alternative parameters." *Ecology*, **52**(4), 577–586.
- Macarthur RH (1965). "Patterns of species diversity." *Biological Reviews*, **40**(4), 510–533.
- Magurran AE (2011). *Measuring biological diversity*, 9 [Nachdr.] edition. Blackwell, Malden, Mass. ISBN 978-0-632-05633-0.
- Margalef R (1973). "Information theory in ecology." *International Journal of General Systems*, **3**, 36–71.
- McIntosh RP (1967). "An index of diversity and the relation of certain concepts to diversity." *Ecology*, **48**(3), 392–404.
- Menhinick EF (1964). "A comparison of some species-individuals diversity indices applied to samples of field insects." *Ecology*, **45**(4), 859–861.
- Nei M (1973). "Analysis of gene diversity in subdivided populations." *Proceedings of the National Academy of Sciences*, **70**(12), 3321–3323.
- Peet RK (1974). "The measurement of species diversity." *Annual Review of Ecology and Systematics*, **5**(1), 285–307.
- Pielou EC (1966). "The measurement of diversity in different types of biological collections." *Journal of Theoretical Biology*, **13**, 131–144.

Pielou EC (1975). *Ecological diversity*. New York : Wiley. ISBN 978-0-471-68925-6.

Renyi A (1960). "On measures of entropy and information." In Neyman J (ed.), *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability (June 20-July 30 1960), Volume I: Contributions to the Theory of Statistics*, 547–561. University of California Press.

Shannon CE, Weaver W (1949). *The Mathematical Theory of Communication*, number v. 2 in *The Mathematical Theory of Communication*. University of Illinois Press.

Simpson EH (1949). "Measurement of diversity." *Nature*, **163**(4148), 688–688.

Smith B, Wilson JB (1996). "A consumer's guide to evenness indices." *Oikos*, **76**(1), 70.

Tsallis C (1988). "Possible generalization of Boltzmann-Gibbs statistics." *Journal of Statistical Physics*, **52**(1-2), 479–487.

Williams CB (1964). *Patterns in the Balance of Nature and Related Problems in Quantitative Ecology*. Academic Press.

Examples

```
#-----
# Qualitative trait data ----
#-----

library(EvaluateCore)

pdata <- cassava_CC

qual <- c("CUAL", "LNCS", "PTLC", "DSTA", "LFRT", "LBTEF", "CBTR", "NMLB",
         "ANGB", "CUAL9M", "LVC9M", "TNPR9M", "PL9M", "STRP", "STRC",
         "PSTR")

# Convert qualitative data columns to factor
pdata[, qual] <- lapply(pdata[, qual], as.factor)

# Get diversity measures
diversity.calc(x = pdata$CUAL)

# Get diversity measures for multiple qualitative traits
div_out1 <- lapply(pdata[, qual], diversity.calc)
do.call(rbind, div_out1)

#-----
# Species abundance data ----
#-----

library(vegan)
data(dune)

abundance_site1 <- unlist(dune[1, ])
```

```
abundance_site1[abundance_site1 != 0]

# Convert to raw counts for use in diversity.calc using rep
abundance_site1_raw <- factor(rep(names(abundance_site1),
                                times = abundance_site1))

# Get diversity measures
diversity.calc(x = abundance_site1_raw)

# Convert multiple site abundance data to raw counts
abundance_site_raw <- apply(dune, 1, function(x) {
  factor(rep(names(x), times = x))
})

# Get diversity measures for multiple sites
div_out2 <- lapply(abundance_site_raw, diversity.calc)
do.call(rbind, div_out2)
```

diversity.compare *Compare Diversity Measures*

Description

The function `diversity.compare` compares diversity indices between different groups using the following approaches.

- Global permutation test across all groups simultaneously.
- Pairwise tests between all combinations of groups.
- Bootstrap confidence intervals (CI) for each group.
- Diversity profiles (Hill, Renyi and Tsallis) for each group.

Usage

```
diversity.compare(
  x,
  group,
  R = 1000,
  base = exp(1),
  na.omit = TRUE,
  global.test = TRUE,
  pairwise.test = TRUE,
  bootstrap.ci = TRUE,
  diversity.profile = TRUE,
  p.adjust.method = c("bonferroni", "holm"),
  ci.conf = 0.95,
  ci.type = c("perc", "bca"),
  q = seq(0, 3, 0.1),
```

```

parallel = c("no", "multicore", "snow"),
ncpus = 1L,
cl = NULL,
seed = 123
)

```

Arguments

<code>x</code>	A factor vector of categories (e.g., species, traits). The frequency of each level is treated as the abundance of that category.
<code>group</code>	A factor vector indicating the group of each observation. Must have the same length as <code>x</code> .
<code>R</code>	Integer specifying the number of permutations. Default is 1000.
<code>base</code>	The logarithm base to be used for computation of shannon family of diversity indices. Default is <code>exp(1)</code> .
<code>na.omit</code>	logical. If TRUE, missing values (NA) are ignored and not included as a distinct factor level for computation. Default is TRUE.
<code>global.test</code>	logical. If TRUE performs the global permutation tests for the diversity measures. Default is TRUE.
<code>pairwise.test</code>	logical. If TRUE performs the pairwise permutation tests for the diversity measures. Default is TRUE.
<code>bootstrap.ci</code>	logical. If TRUE computes the bootstrap confidence intervals for the diversity measures. Default is TRUE.
<code>diversity.profile</code>	logical. If TRUE diversity profiles. Default is TRUE.
<code>p.adjust.method</code>	(<code>perm.test.pairwise</code> only) Method for adjusting p-values for multiple comparisons. Options include "bonferroni" and "holm". Default is "bonferroni".
<code>ci.conf</code>	Confidence level of the bootstrap interval. Default is 0.95.
<code>ci.type</code>	A vector of character strings representing the type of intervals required. The options are <code>c("perc", "bca")</code> .
<code>q</code>	The order of the parametric index.
<code>parallel</code>	The type of parallel operation to be used (if any). If missing, the default is taken from the option "boot.parallel" (and if that is not set, "no").
<code>ncpus</code>	integer: number of processes to be used in parallel operation: typically one would chose this to the number of available CPUs.
<code>cl</code>	An optional parallel or snow cluster for use if <code>parallel = "snow"</code> . If not supplied, a cluster on the local machine is created for the duration of the boot call.
<code>seed</code>	Integer. Random seed used to ensure reproducibility of permutations and bootstrap. Default is 123.

Value

A list with the following elements.

Diversity Indices A data frame of the different diversity indices computed for each group.

Global Test A data frame of results of global permutation test including the test statistic (weighted sum of squares between group summary indices) and the p value for the different diversity indices.

Pairwise Test A list of the following data frames.

p-value A data frame of p values for each between group comparison for different diversity measures.

cld A data frame of compact letter displays of significant differences among groups for different diversity measures.

Bootstrap CIs A data frame of lower and upper bootstrap confidence intervals computed for each group in different diversity measures.

Diversity profiles A list of data frames of Hill, Renyi and Tsallis diversity profiles computed for each group.

Note

- In small samples with bounded statistics like Shannon Diversity Index and Menhinick index, the bootstrap upper CI can equal the observed value because resamples cannot exceed the theoretical maximum.
- Similarly in small samples, the lower confidence bound can be zero because bootstrap resamples occasionally can contain only a single category (class or species), due to sampling uncertainty and the natural lower bound of the diversity index like Shannon Diversity Index.
- The BCa bootstrap can produce negative lower confidence limits due to boundary effects and skewness in the resampled distribution.

Examples

```
library(EvaluateCore)

pdata <- cassava_CC

qual <- c("CUAL", "LNCS", "PTLC", "DSTA", "LFRT", "LBTEF", "CBTR", "NMLB",
         "ANGB", "CUAL9M", "LVC9M", "TNPR9M", "PL9M", "STRP", "STRC",
         "PSTR")

# Convert qualitative data columns to factor
pdata[, qual] <- lapply(pdata[, qual], as.factor)

str(pdata)

diversity.compare(x = pdata$CUAL, group = pdata$LNCS, R = 100,
                 base = exp(1), na.omit = TRUE)

diversity.compare(x = pdata$ANGB, group = pdata$LNCS, R = 100,
                 base = exp(1), na.omit = TRUE)
```

diversity.profile
Generate Diversity Profiles for Parametric Indices

Description

Computes diversity profiles across a continuous range of sensitivity parameter (The order q) using parametric diversity indices such as Hill numbers, Rényi entropy, and Tsallis entropy. The function also supports the generation of multiple profiles to enable comparisons among groups. It provides flexible options for generation of bootstrap confidence intervals for the values.

Usage

```
diversity.profile(
  x,
  group,
  q = seq(0, 3, 0.1),
  na.omit = TRUE,
  ci.conf = 0.95,
  R = 1000,
  parameter = c("hill", "renyi", "tsallis"),
  ci.type = c("perc", "bca"),
  parallel = c("no", "multicore", "snow"),
  ncpus = getOption("boot.ncpus", 1L),
  cl = NULL,
  seed = 123
)
```

Arguments

<code>x</code>	A numeric or factor vector of observations.
<code>group</code>	A factor vector indicating the group of each observation. Must have the same length as <code>x</code> .
<code>q</code>	The order of the parametric index.
<code>na.omit</code>	logical. If TRUE, missing values (NA) are ignored and not included as a distinct factor level for computation. Default is TRUE.
<code>ci.conf</code>	Confidence level of the bootstrap interval. Default is 0.95.
<code>R</code>	Integer specifying the number of permutations. Default is 1000.
<code>parameter</code>	The parametric index. Options include "hill", "renyi" and "tsallis". Default is "hill".
<code>ci.type</code>	A vector of character strings representing the type of intervals required. The options are <code>c("perc", "bca")</code> .
<code>parallel</code>	The type of parallel operation to be used (if any). If missing, the default is taken from the option "boot.parallel" (and if that is not set, "no").
<code>ncpus</code>	integer: number of processes to be used in parallel operation: typically one would chose this to the number of available CPUs.

<code>cl</code>	An optional parallel or snow cluster for use if <code>parallel = "snow"</code> . If not supplied, a cluster on the local machine is created for the duration of the boot call.
<code>seed</code>	Integer. Random seed used to ensure reproducibility of bootstrap. Default is 123.

Details

See **Parametric Indices** in [diversity.calc](#) for theoretical details.

Value

A list of data frames with the following columns for each factor level in group.

q
observed
mean
lower
upper

Note

The default number of bootstrap replicates $R = 1000$ is provided for quick exploratory analysis. For more reliable (but slower) confidence intervals or standard error estimates it is strongly recommended to increase R to at 5000-10000, depending on your data and required precision.

Examples

```
library(EvaluateCore)
library(dplyr)
library(ggplot2)

pdata <- cassava_CC

qual <- c("CUAL", "LNGS", "PTLC", "DSTA", "LFRT", "LBTEF", "CBTR", "NMLB",
         "ANGB", "CUAL9M", "LVC9M", "TNPR9M", "PL9M", "STRP", "STRC",
         "PSTR")

# Convert qualitative data columns to factor
pdata[, qual] <- lapply(pdata[, qual], as.factor)

str(pdata)

important_q <- c(0, 1, 2)
important_labels <- c("0D", "1D", "2D")

# NOTE: Increase R to 10000 for more reliable (but slower) estimates.

# Hill profile - Percentile CIs ----

hill_profile1 <-
```

```

diversity.profile(x = pdata$CUAL, group = pdata$LNKS,
                 parameter = "hill", ci.type = "perc",
                 R = 100)
hill_profile1

hill_profile1_df <- dplyr::bind_rows(hill_profile1, .id = "group")

hill_points1_df <- hill_profile1_df %>%
  filter(q %in% important_q) %>%
  mutate(order_label = factor(q, levels = important_q,
                              labels = important_labels))

ggplot(hill_profile1_df, aes(x = q, y = observed,
                           color = group, fill = group)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2, color = NA) +
  geom_line(linewidth = 1) +
  geom_vline(xintercept = c(0, 1, 2), linetype = "dashed",
            color = "grey60") +
  geom_point(data = hill_points1_df, aes(shape = order_label),
            size = 3, stroke = 1, inherit.aes = TRUE) +
  scale_shape_manual(values = c(17, 19, 15), name = "Important q") +
  labs(x = "Order (q)", y = "Hill number",
       color = "Group", fill = "Group") +
  theme_bw()

ggplot(hill_profile1_df, aes(x = q, y = observed)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), fill = "grey80") +
  geom_line(color = "black", linewidth = 1) +
  facet_wrap(~ group, scales = "free_y") +
  labs(x = "Order (q)", y = "Hill number") +
  theme_bw()

# Rényi profile - Percentile CIs ----

renyi_profile1 <-
  diversity.profile(pdata$CUAL, group = pdata$LNKS,
                  parameter = "renyi", ci.type = "perc",
                  R = 100)
renyi_profile1

renyi_profile1_df <- dplyr::bind_rows(renyi_profile1, .id = "group")

renyi_points1_df <- renyi_profile1_df %>%
  filter(q %in% important_q) %>%
  mutate(order_label = factor(q, levels = important_q,
                              labels = important_labels))

ggplot(renyi_profile1_df, aes(x = q, y = observed,
                             color = group, fill = group)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2, color = NA) +
  geom_line(linewidth = 1) +
  geom_vline(xintercept = c(0, 1, 2), linetype = "dashed",
            color = "grey60") +

```

```

geom_point(data = renyi_points1_df, aes(shape = order_label),
           size = 3, stroke = 1, inherit.aes = TRUE) +
scale_shape_manual(values = c(17, 19, 15), name = "Important q") +
labs(x = "Order (q)", y = "Hill number",
     color = "Group", fill = "Group") +
theme_bw()

ggplot(renyi_profile1_df, aes(x = q, y = observed)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), fill = "grey80") +
  geom_line(color = "black", linewidth = 1) +
  facet_wrap(~ group, scales = "free_y") +
  labs(x = "Order (q)", y = "Hill number") +
  theme_bw()

# Tsallis profile - Percentile CIs ----

tsallis_profile1 <-
  diversity.profile(pdata$CUAL, group = pdata$LNGS,
                  parameter = "tsallis", ci.type = "perc",
                  R = 100)
tsallis_profile1

tsallis_profile1_df <- dplyr::bind_rows(tsallis_profile1, .id = "group")

tsallis_points1_df <- tsallis_profile1_df %>%
  filter(q %in% important_q) %>%
  mutate(order_label = factor(q, levels = important_q,
                              labels = important_labels))

ggplot(tsallis_profile1_df, aes(x = q, y = observed,
                              color = group, fill = group)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2, color = NA) +
  geom_line(linewidth = 1) +
  geom_vline(xintercept = c(0, 1, 2), linetype = "dashed",
            color = "grey60") +
  geom_point(data = tsallis_points1_df, aes(shape = order_label),
            size = 3, stroke = 1, inherit.aes = TRUE) +
  scale_shape_manual(values = c(17, 19, 15), name = "Important q") +
  labs(x = "Order (q)", y = "Hill number",
     color = "Group", fill = "Group") +
  theme_bw()

ggplot(tsallis_profile1_df, aes(x = q, y = observed)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), fill = "grey80") +
  geom_line(color = "black", linewidth = 1) +
  facet_wrap(~ group, scales = "free_y") +
  labs(x = "Order (q)", y = "Hill number") +
  theme_bw()

# Hill profile - BCa CIs ----

hill_profile2 <-
  diversity.profile(pdata$CUAL, group = pdata$LNGS,

```

```

        parameter = "hill", ci.type = "bca",
        R = 100)
hill_profile2

hill_profile2_df <- dplyr::bind_rows(hill_profile2, .id = "group")

hill_points2_df <- hill_profile2_df %>%
  filter(q %in% important_q) %>%
  mutate(order_label = factor(q, levels = important_q,
                              labels = important_labels))

ggplot(hill_profile2_df, aes(x = q, y = observed,
                           color = group, fill = group)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2, color = NA) +
  geom_line(linewidth = 1) +
  geom_vline(xintercept = c(0, 1, 2), linetype = "dashed",
            color = "grey60") +
  geom_point(data = hill_points2_df, aes(shape = order_label),
            size = 3, stroke = 1, inherit.aes = TRUE) +
  scale_shape_manual(values = c(17, 19, 15), name = "Important q") +
  labs(x = "Order (q)", y = "Hill number",
       color = "Group", fill = "Group") +
  theme_bw()

ggplot(hill_profile2_df, aes(x = q, y = observed)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), fill = "grey80") +
  geom_line(color = "black", linewidth = 1) +
  facet_wrap(~ group, scales = "free_y") +
  labs(x = "Order (q)", y = "Hill number") +
  theme_bw()

# Rényi profile - BCa CIs ----

renyi_profile2 <-
  diversity.profile(pdata$CUAL, group = pdata$LNCS,
                  parameter = "renyi", ci.type = "bca",
                  R = 100)
renyi_profile2

renyi_profile2_df <- dplyr::bind_rows(renyi_profile2, .id = "group")

renyi_points2_df <- renyi_profile2_df %>%
  filter(q %in% important_q) %>%
  mutate(order_label = factor(q, levels = important_q,
                              labels = important_labels))

ggplot(renyi_profile2_df, aes(x = q, y = observed,
                             color = group, fill = group)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2, color = NA) +
  geom_line(linewidth = 1) +
  geom_vline(xintercept = c(0, 1, 2), linetype = "dashed",
            color = "grey60") +
  geom_point(data = renyi_points2_df, aes(shape = order_label),

```

```

      size = 3, stroke = 1, inherit.aes = TRUE) +
    scale_shape_manual(values = c(17, 19, 15), name = "Important q") +
    labs(x = "Order (q)", y = "Hill number",
         color = "Group", fill = "Group") +
    theme_bw()

ggplot(renyi_profile2_df, aes(x = q, y = observed)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), fill = "grey80") +
  geom_line(color = "black", linewidth = 1) +
  facet_wrap(~ group, scales = "free_y") +
  labs(x = "Order (q)", y = "Hill number") +
  theme_bw()

# Tsallis profile - BCa CIs ----

tsallis_profile2 <-
  diversity.profile(pdata$CUAL, group = pdata$LNGS,
                   parameter = "tsallis", ci.type = "bca",
                   R = 100)

tsallis_profile2

tsallis_profile2_df <- dplyr::bind_rows(tsallis_profile2, .id = "group")

tsallis_points2_df <- tsallis_profile2_df %>%
  filter(q %in% important_q) %>%
  mutate(order_label = factor(q, levels = important_q,
                              labels = important_labels))

ggplot(tsallis_profile2_df, aes(x = q, y = observed,
                              color = group, fill = group)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2, color = NA) +
  geom_line(linewidth = 1) +
  geom_vline(xintercept = c(0, 1, 2), linetype = "dashed",
            color = "grey60") +
  geom_point(data = tsallis_points2_df, aes(shape = order_label),
            size = 3, stroke = 1, inherit.aes = TRUE) +
  scale_shape_manual(values = c(17, 19, 15), name = "Important q") +
  labs(x = "Order (q)", y = "Hill number",
       color = "Group", fill = "Group") +
  theme_bw()

ggplot(tsallis_profile2_df, aes(x = q, y = observed)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), fill = "grey80") +
  geom_line(color = "black", linewidth = 1) +
  facet_wrap(~ group, scales = "free_y") +
  labs(x = "Order (q)", y = "Hill number") +
  theme_bw()

```

Description

These are core low-level routines for the computation of multiple diversity indices, designed to be used by `diversity.calc` and other functions.

Usage

```
berger_parker(x)  
berger_parker_reciprocal(x)  
simpson(x)  
gini_simpson(x)  
simpson_max(x)  
simpson_reciprocal(x)  
simpson_relative(x)  
simpson_evenness(x)  
shannon(x, base = 2)  
shannon_max(x, base = 2)  
shannon_relative(x, base = 2)  
shannon_ens(x, base = 2)  
mcintosh_diversity(x)  
mcintosh_evenness(x)  
smith_wilson(x, warn = TRUE)  
heip_evenness(x)  
margalef_index(x)  
menhinick_index(x)  
brillouin_index(x)  
hill_number(x, q = 1)  
renyi_entropy(x, q = 1)
```

```
tsallis_entropy(x, q = 1)
```

```
hill_evenness(x, q = 1)
```

Arguments

x	A factor vector of categories (e.g., species, traits). The frequency of each level is treated as the abundance of that category.
base	The logarithm base to be used for computation of shannon family of diversity indices. Default is exp(1).
warn	logical. If TRUE shows the relevant warning. Default is TRUE.
q	The order of the parametric index.

Value

The calculated diversity index value.

permutation_tests *Permutation Tests*

Description

These functions perform permutation-based hypothesis tests to compare groups with respect to a summary statistic (e.g., mean, diversity index).

- `perm.test.global` performs a global test across all groups simultaneously, using a weighted sum of squares between group summary indices as the test statistic
- `perm.test.pairwise` performs pairwise tests between all combinations of groups, comparing the absolute difference in summary statistics and optionally adjusting p-values for multiple comparisons.

Usage

```
perm.test.global(  
  x,  
  group,  
  fun,  
  R = 1000,  
  na.omit = TRUE,  
  fun.args = list(),  
  max.invalid = 0.25,  
  parallel = c("no", "multicore", "snow"),  
  ncpus = 1L,  
  cl = NULL,  
  seed = 123  
)
```

```
perm.test.pairwise(
  x,
  group,
  fun,
  R = 1000,
  na.omit = TRUE,
  fun.args = list(),
  p.adjust.method = c("bonferroni", "holm"),
  max.invalid = 0.25,
  parallel = c("no", "multicore", "snow"),
  ncpus = 1L,
  cl = NULL,
  seed = 123
)
```

Arguments

<code>x</code>	A numeric or factor vector of observations.
<code>group</code>	A factor vector indicating the group of each observation. Must have the same length as <code>x</code> .
<code>fun</code>	A function to summarize values within each group.
<code>R</code>	Integer specifying the number of permutations. Default is 1000.
<code>na.omit</code>	logical. If TRUE, when <code>x</code> is a factor, missing values (NA) are ignored and not included as a distinct factor level for computation. Default is TRUE.
<code>fun.args</code>	Named list of additional arguments forwarded to 'fun'.
<code>max.invalid</code>	Numeric between 0 and 1. Maximum allowed proportion of invalid permutations (i.e., permutations for which the test statistic is non-finite). If the proportion of invalid permutations exceeds this threshold, the function stops execution with an error, indicating that the statistic function is not permutation-stable.
<code>parallel</code>	The type of parallel operation to be used (if any). If missing, the default is taken from the option "boot.parallel" (and if that is not set, "no").
<code>ncpus</code>	integer: number of processes to be used in parallel operation: typically one would chose this to the number of available CPUs.
<code>cl</code>	An optional parallel or snow cluster for use if <code>parallel = "snow"</code> . If not supplied, a cluster on the local machine is created for the duration of the boot call.
<code>seed</code>	Integer. Random seed used to ensure reproducibility of permutations. Default is 123.
<code>p.adjust.method</code>	(perm.test.pairwise only) Method for adjusting p-values for multiple comparisons. Options include "bonferroni" and "holm". Default is "bonferroni".

Value

`perm.test.global` A list of the following elements.

- test_stat** The test statistic value.

observed_values The observed values for each group.

p_value The p value.

perm.test.pairwise A data frame of the following columns.

Comparison The comparison.

p.value The p value.

adj.p.value The adjusted p value.

Examples

```
library(EvaluateCore)

pdata <- cassava_CC

qual <- c("CUAL", "LNGS", "PTLC", "DSTA", "LFRT", "LBTEF", "CBTR", "NMLB",
         "ANGB", "CUAL9M", "LVC9M", "TNPR9M", "PL9M", "STRP", "STRC",
         "PSTR")

# Conver '#t qualitative data columns to factor
pdata[, qual] <- lapply(pdata[, qual], as.factor)

str(pdata)

# NOTE: Increase R to 10000 for more reliable (but slower) estimates.

# Global tests ----

perm.test.global(x = pdata$NMSR, group = pdata$CUAL, fun = mean,
                R = 100)

perm.test.global(x = pdata$LNGS, group = pdata$CUAL, fun = shannon,
                R = 100)

perm.test.global(x = pdata$PTLC, group = pdata$CUAL, fun = simpson,
                R = 100)

# Pairwise tests ----

perm.test.pairwise(x = pdata$NMSR, group = pdata$CUAL, fun = mean,
                  R = 100)

perm.test.pairwise(x = pdata$LNGS, group = pdata$CUAL, fun = shannon,
                  R = 100)

perm.test.pairwise(x = pdata$PTLC, group = pdata$CUAL, fun = simpson,
                  R = 100)
```

Index

berger_parker (diversity_functions), 21
berger_parker_reciprocal
 (diversity_functions), 21
boot, 2
boot.ci, 2
bootstrap.ci, 2
brillouin_index (diversity_functions),
 21

diversity.calc, 5, 17, 22
diversity.compare, 13
diversity.profile, 16
diversity_functions, 21

gini_simpson (diversity_functions), 21

heip_evenness (diversity_functions), 21
hill_evenness (diversity_functions), 21
hill_number (diversity_functions), 21

margalef_index (diversity_functions), 21
mcintosh_diversity
 (diversity_functions), 21
mcintosh_evenness
 (diversity_functions), 21
menhinick_index (diversity_functions),
 21

perm.test.global (permutation_tests), 23
perm.test.pairwise (permutation_tests),
 23
permutation_tests, 23

renyi_entropy (diversity_functions), 21

shannon (diversity_functions), 21
shannon_ens (diversity_functions), 21
shannon_max (diversity_functions), 21
shannon_relative (diversity_functions),
 21
simpson (diversity_functions), 21
simpson_evenness (diversity_functions),
 21
simpson_max (diversity_functions), 21
simpson_reciprocal
 (diversity_functions), 21
simpson_relative (diversity_functions),
 21
smith_wilson (diversity_functions), 21
tsallis_entropy (diversity_functions),
 21