

# Package ‘entropart’

July 2, 2014

**Type** Package

**Title** Entropy partitioning to measure diversity

**Version** 1.2.0

**Date** 2014-05-16

**Author** Eric Marcon, Bruno Hérault

**Maintainer** Eric Marcon <Eric.Marcon@ecofog.gf>

**Description** Measurement and partitioning of diversity, based on Tsallis entropy.

**License** GNU General Public License

**Imports** ade4

**Suggests** vegan

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-05-17 07:39:54

## R topics documented:

entropart-package . . . . .	3
AllenH . . . . .	4
AlphaDiversity . . . . .	6
AlphaEntropy . . . . .	7
BetaDiversity . . . . .	8
BetaEntropy . . . . .	10
ChaoPD . . . . .	11
CommunityProfile . . . . .	13
Coverage . . . . .	14
Diversity . . . . .	15
DivEst . . . . .	16

DivPart	18
DivProfile	21
Dqz	23
EightSpAbundance	25
EightSpTree	25
Enq	26
EntropyCI	27
expq	28
GammaDiversity	29
GammaEntropy	30
Hqz	32
HqzBeta	33
KLq	35
lnq	36
MC Utilities	37
MCdiversity	38
MCentropy	40
MetaCommunity	41
Originality.Species	43
Paracou618.dist	44
Paracou618.Functional	45
Paracou618.MC	46
Paracou618.Taxonomy	46
PDFD	47
PhyloApply	48
PhyloBetaEntropy	50
PhyloDiversity	52
PhyloEntropy	54
plot.PhyloValue	56
PPtree	57
Rao	58
Richness	59
Shannon	60
ShannonBeta	62
Simpson	63
SimpsonBeta	64
SimTest	66
Tsallis	67
TsallisBeta	69

---

entropart-package	<i>Entropy partitioning to measure diversity</i>
-------------------	--------------------------------------------------

---

## Description

This package provides a full set of functions to calculate alpha, beta and gamma diversity of communities, including phylogenetic diversity.

Estimation-bias corrections are available.

## Details

Package: entropart  
 Type: Package  
 Version: 1.2.0  
 Date: 2014-05-16  
 License: GNU General Public License

In the entropart package, individuals of different "species" are counted in several "communities" which are aggregated to define a "metacommunity". In the metacommunity, the probability to find a species in the weighted average of probabilities in communities.

This is a naming convention, which may correspond to plots in a forest inventory or any data organized the same way.

The simplest way to import data is to organize it into two text files. The first file should contain abundance data: the first column named Species for species names, and a column for each community.

Species	NameOfCommunity1	NameOfCommunity2
NameOfSpecies1	1	5
NameOfSpecies2	4	2
...	...	...

The second file should contain the community weights and be organized as follows:

Communities	Weights
NameOfCommunity1	3
NameOfCommunity2	1

Files can be read and data imported by code such as:

```
Abundances <- read.csv(file="Abundances.csv")
Weights <- read.csv(file="Weights.csv")
MC <- MetaCommunity(Abundances, Weights)
```

The last line of the code calls the `MetaCommunity` function to create an object that will be used by all metacommunity functions, such as `DivPart` (to partition diversity), `DivEst` (to partition diversity and calculate confidence interval of its estimation) or `DivProfile` (to compute diversity profiles).

The package also provides community functions to calculate entropy or diversity with bias correction from a vector of probabilities or abundances, such as `bcShannon` (to calculate the unbiased estimator of Shannon entropy of a community) or `Diversity` (to calculate the true diversity of a community).

Phylogenetic entropy and diversity can be calculated if a phylogenetic (or functional), ultrametric tree is provided. See `PhyloEntropy`, `Rao` for examples of entropy and `bcPhyloDiversity` to calculate phylodiversity, with bias correction. Similarity-based diversity is calculated with `Dqz` or (with bias correction) `bcDqz`, based on a similarity matrix.

### Author(s)

Eric Marcon, Bruno Herault

### References

- Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).
- Marcon, E. and Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).
- Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.
- Marcon, E., Herault, B., Baraloto, C. and Lang, G. (2012). The Decomposition of Shannon's Entropy and a Confidence Interval for Beta Diversity. *Oikos* 121(4): 516-522.

---

AllenH

*Phylogenetic Entropy of a Community*

---

### Description

Calculates the phylogenetic diversity of order  $q$  of a probability vector.

### Usage

```
AllenH(Ps, q = 1, PhyloTree, Normalize = TRUE, CheckArguments = TRUE)
```

### Arguments

- |           |                                                                                                                                                                                               |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ps        | A probability vector, summing to 1.                                                                                                                                                           |
| q         | A number: the order of entropy. Default is 1.                                                                                                                                                 |
| PhyloTree | An object of class <code>hclust</code> or <code>phylog</code> . The tree is not necessarily ultrametric. Computation is must faster if the tree is <code>phylog</code> or <code>PPtree</code> |
| Normalize | If TRUE (default), diversity is not affected by the height of the tree. If FALSE, it is proportional to the height of the tree.                                                               |

**CheckArguments** Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

### Details

The phylogenetic entropy is calculated following Allen *et al.* (2009) for order  $q = 1$  and Leinster and Cobold (2011) for other orders. The result is identical to the total entropy calculated by [PhyloEntropy](#) but it is much faster. A single value is returned instead of a [PhyloEntropy](#) object, and no bias correction is available.

The `Normalize` argument allows normalizing entropy by the height of the tree, similarly to [ChaoPD](#).

This function allows very efficient calculation when the Tree is of class [phylog](#) or [PPtree](#) ([hclust](#) trees must be converted to [phylog](#), a slow process).

Diversity can be calculated for non ultrametric trees following Leinster and Cobold (2011) even though the meaning of the result is not so clear.

### Value

A number equal the entropy of the community.

### Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf>

### References

Allen, B., Kon, M. and Bar-Yam, Y. (2009). A New Phylogenetic Diversity Measure Generalizing the Shannon Index and Its Application to Phyllostomid Bats. *American Naturalist* 174(2): 236-243.

Leinster, T. and Cobbold, C. (2011). Measuring diversity: the importance of species similarity. *Ecology* 93(3): 477-489.

### See Also

[PhyloEntropy](#), [ChaoPD](#)

### Examples

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest
# and their taxonomy)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps

# Calculate the phylogenetic Shannon diversity of the plot
AllenH(Paracou618.MC$Ps, 1, Paracou618.Taxonomy, Normalize=TRUE) -> Allen
Allen

# Calculate it using PhyloEntropy: more powerful but much slower is the tree has many periods
PhyloEntropy(Paracou618.MC$Ps, 1, Paracou618.Taxonomy, Normalize=TRUE) -> phyE
summary(phyE)
```

---

AlphaDiversity	<i>Unbiased alpha diversity of a metacommunity</i>
----------------	----------------------------------------------------

---

### Description

Calculates the unbiased total alpha diversity of order  $q$  of communities.

### Usage

```
AlphaDiversity(MC, q = 1, Correction = "Best", Tree = NULL, Normalize = TRUE,
  Z = NULL, CheckArguments = TRUE)
```

### Arguments

MC	A <a href="#">MetaCommunity</a> object.
q	A number: the order of diversity. Default is 1 for Shannon diversity.
Correction	A string containing one of the possible corrections accepted by <a href="#">AlphaEntropy</a> or "None" or "Best", the default value.
Tree	An object of class <a href="#">hclust</a> or <a href="#">phylog</a> . The tree must be ultrametric.
Normalize	If TRUE (default), diversity is not affected by the height of the tree. If FALSE, diversity is proportional to the height of the tree.
Z	A relatedness matrix, <i>i.e.</i> a square matrix whose terms are all positive, strictly positive on the diagonal. Generally, the matrix is a similarity matrix, <i>i.e.</i> the diagonal terms equal 1 and other terms are between 0 and 1.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

### Details

Entropy is calculated by [AlphaEntropy](#) and transformed into diversity.

### Value

An [MCdiversity](#) object containing diversity values of each community and of the metacommunity.

### Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf>

### References

- Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.
- Marcon, E., Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).
- Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).

**See Also**[AlphaEntropy](#)**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Calculate Simpson alpha diversity
summary(AlphaDiversity(Paracou618.MC, 2))
# Compare without correction
summary(AlphaDiversity(Paracou618.MC, 2, Correction = "None"))
# Estimate phylogenetic Simpson alpha diversity
summary(AlphaDiversity(Paracou618.MC, 2, Tree = Paracou618.Taxonomy) -> e)
plot(e)
```

AlphaEntropy

*Unbiased alpha entropy of a metacommunity***Description**

Calculates the unbiased total alpha entropy of order  $q$  of communities.

**Usage**

```
AlphaEntropy(MC, q = 1, Correction = "Best", Tree = NULL, Normalize = TRUE,
             Z = NULL, CheckArguments = TRUE)
```

**Arguments**

MC	A <a href="#">MetaCommunity</a> object.
q	A number: the order of diversity. Default is 1 for Shannon entropy.
Correction	A string containing one of the possible corrections accepted by the bias-corrected entropy function (see details) or "None" or "Best", the default value.
Tree	An object of class <a href="#">hclust</a> or <a href="#">phylog</a> . The tree must be ultrametric.
Normalize	If TRUE (default), the entropy returned by the function is normalized by the height of the tree (it is the weighted average value of the entropy in each slice). If FALSE, it is the unnormalized weighted sum of the results.
Z	A relatedness matrix, <i>i.e.</i> a square matrix whose terms are all positive, strictly positive on the diagonal. Generally, the matrix is a similarity matrix, <i>i.e.</i> the diagonal terms equal 1 and other terms are between 0 and 1.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

If Tree is not NULL, then phylogenetic entropy is calculated by [bcPhyloEntropy](#) else if Z is not NULL, then similarity-based entropy is calculated by [bcHqz](#), else neutral entropy is calculated by [bcTsallis](#), else .

The alpha entropy of each community is calculated and summed according to community weights.

**Value**

An [MEntropy](#) object containing entropy values of each community and of the metacommunity.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.

Marcon, E., Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).

Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).

**See Also**

[bcTsallis](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Calculate Simpson alpha entropy
summary(AlphaEntropy(Paracou618.MC, 2))
# Compare without correction
summary(AlphaEntropy(Paracou618.MC, 2, Correction = "None"))
# Estimate phylogenetic Simpson alpha entropy
summary(AlphaEntropy(Paracou618.MC, 2, Tree = Paracou618.Taxonomy) -> e)
plot(e)
```

---

BetaDiversity

*Unbiased beta diversity of a metacommunity*

---

**Description**

Calculates the unbiased beta diversity of order  $q$  between communities.



**Usage**

```
BetaDiversity(MC, q = 1, Correction = "Best", Tree = NULL, Normalize = TRUE,
              Z = NULL, CheckArguments = TRUE)
```

**Arguments**

MC	A <a href="#">MetaCommunity</a> object.
q	A number: the order of diversity. Default is 1 for Shannon diversity.
Correction	A string containing one of the possible corrections accepted by <a href="#">bcTsallisBeta</a> or "None" or "Best", the default value.
Tree	An object of class <a href="#">hclust</a> or <a href="#">phylog</a> . The tree must be ultrametric.
Normalize	If TRUE (default), diversity is not affected by the height of the tree. If FALSE, diversity is proportional to the height of the tree.
Z	A relatedness matrix, <i>i.e.</i> a square matrix whose terms are all positive, strictly positive on the diagonal. Generally, the matrix is a similarity matrix, <i>i.e.</i> the diagonal terms equal 1 and other terms are between 0 and 1.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

Entropy is calculated by [BetaEntropy](#) and transformed into diversity.

Diversity values of communities are not defined: community entropies are averaged to obtain the metacommunity entropy which is transformed into diversity (Marcon et al., 2014).

**Value**

An [MCdiversity](#) object containing diversity value of the metacommunity.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

- Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.
- Marcon, E., Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).
- Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).

**See Also**

[BetaEntropy](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Estimate Shannon beta diversity
summary(BetaDiversity(Paracou618.MC, 1))
# Compare without correction
summary(BetaDiversity(Paracou618.MC, 1, Correction = "None"))
# Estimate phylogenetic Shannon beta diversity
summary(BetaDiversity(Paracou618.MC, 1, Tree = Paracou618.Taxonomy) -> e)
```

BetaEntropy

*Unbiased beta entropy of a metacommunity***Description**

Calculates the unbiased beta entropy of order  $q$  between communities.

**Usage**

```
BetaEntropy(MC, q = 1, Correction = "Best", Tree = NULL, Normalize = TRUE,
  Z = NULL, CheckArguments = TRUE)
```

**Arguments**

MC	A <a href="#">MetaCommunity</a> object.
q	A number: the order of diversity. Default is 1 for Shannon entropy.
Correction	A string containing one of the possible corrections accepted by the bias-corrected entropy function (see details) or "None" or "Best", the default value.
Tree	An object of class <a href="#">hclust</a> or <a href="#">phylog</a> . The tree must be ultrametric.
Normalize	If TRUE (default), the entropy returned by the function is normalized by the height of the tree (it is the weighted average value of the entropy in each slice). If FALSE, it is the unnormalized weighted sum of the results.
Z	A relatedness matrix, <i>i.e.</i> a square matrix whose terms are all positive, strictly positive on the diagonal. Generally, the matrix is a similarity matrix, <i>i.e.</i> the diagonal terms equal 1 and other terms are between 0 and 1.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

If Tree is not NULL, then phylogenetic entropy is calculated by [bcPhyloBetaEntropy](#) else if Z is not NULL, then similarity-based entropy is calculated by [bcHqzBeta](#), else neutral entropy is calculated by [bcTsallisBeta](#), else .

The unbiased beta entropy of each community is calculated and summed according to community weights.

Note that beta entropy is related to alpha entropy (if  $q$  is not 1) and cannot be compared across communities (Jost, 2007). Do rather calculate the [BetaDiversity](#) of the metacommunity.

**Value**

An [MEntropy](#) object containing entropy values of each community and of the metacommunity.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.

Marcon, E., Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).

Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).

**See Also**

[bcTsallisBeta](#), [BetaDiversity](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Estimate Shannon beta entropy
summary(BetaEntropy(Paracou618.MC, 1))
# Compare without correction
summary(BetaEntropy(Paracou618.MC, 1, Correction = "None"))
# Estimate phylogenetic Shannon beta entropy
summary(BetaEntropy(Paracou618.MC, 1, Tree = Paracou618.Taxonomy) -> e)
plot(e)
```

---

ChaoPD

*Phylogenetic Diversity of a Community*

---

**Description**

Calculates the phylogenetic diversity of order  $q$  of a probability vector.

**Usage**

```
ChaoPD(Ps, q = 1, PhyloTree, Normalize = TRUE, CheckArguments = TRUE)
```

**Arguments**

Ps	A probability vector, summing to 1.
q	A number: the order of diversity. Default is 1.
PhyloTree	An object of class <a href="#">hclust</a> or <a href="#">phylog</a> . The tree is not necessarily ultrametric. Computation is much faster if the tree is <a href="#">phylog</a> or <a href="#">PPtree</a>
Normalize	If TRUE (default), diversity is not affected by the height of the tree. If FALSE, it is proportional to the height of the tree.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

The phylogenetic diversity is calculated following Chao *et al.* (2010). The result is identical to the total diversity calculated by [PhyloDiversity](#) but it is much faster. A single value is returned instead of a [PhyloDiversity](#) object, and no bias correction is available.

The Normalize arguments allows calculating either  ${}^q\bar{D}(T)$  (if TRUE) or  ${}^qPD(T)$  if FALSE.

This function allows very efficient calculation when the Tree is of class [phylog](#) or [PPtree](#) ([hclust](#) trees must be converted to [phylog](#), a slow process).

Diversity can be calculated for non ultrametric trees following Chao *et al.* (2010) even though the meaning of the result is not so clear (Leinster and Cobold, 2011).

**Value**

A number equal the diversity of the community.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Chao, A., Chiu, C.-H. and Jost, L. (2010). Phylogenetic diversity measures based on Hill numbers. *Philosophical Transactions of the Royal Society B* 365(1558): 3599-609.

Leinster, T. and Cobbold, C. (2011). Measuring diversity: the importance of species similarity. *Ecology* 93(3): 477-489.

**See Also**

[PhyloDiversity](#), [AllenH](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest
# and their taxonomy)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
```

```
# Calculate the phylogenetic Simpson diversity of the plot
ChaoPD(Paracou618.MC$Ps, 2, Paracou618.Taxonomy, Normalize=TRUE) -> ChaoD
ChaoD

# Calculate it using PhyloDiversity: more powerful but much slower is the tree has many periods
PhyloDiversity(Paracou618.MC$Ps, 2, Paracou618.Taxonomy, Normalize=TRUE) -> phyD
summary(phyD)
```

---

CommunityProfile      *Diversity or Entropy Profile of a community*

---

### Description

Calculates the diversity or entropy profile of a community, applying a community function to a vector of orders.

### Usage

```
CommunityProfile(FUN, NorP, q.seq = seq(0, 2, 0.1), ..., CheckArguments = TRUE)
```

### Arguments

FUN	The function to be applied to each value of $q$ .
NorP	A numeric vector or a two-column matrix. Contains either abundances or probabilities. Two-column matrices should contain the observed abundances (or probabilities) in the first column and the expected ones in the second column, to allow using beta diversity functions.
q.seq	A numeric vector: the sequence of diversity orders to address. Default is from 0 to 2.
...	Additional arguments to be passed to FUN.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

### Details

This function is used to calculate diversity or entropy profiles based on community functions such as [Tsallis](#) or [ChaoPD](#). The first two arguments of the function must be a probability or abundance vector or a two-column matrix (for beta diversity functions) and a number ( $q$ ). Additional arguments cannot be checked. Unexpected results may be returned if FUN is not used properly.

### Value

A list:

x	The order $q$ values
y	The entropy or diversity values returned by FUN

The list can be plotted.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Estimate diversity.
Profile <- CommunityProfile(bcDiversity, Paracou618.MC$Ns, seq(0, 2, 0.1))
plot(Profile, type="l", main="Paracou Plots Diversity", xlab="q", ylab="Diversity")
# Compare to non bias-corrected diversity
lines(CommunityProfile(Diversity, Paracou618.MC$Ps, seq(0, 2, 0.1)), lty=3)
legend("topright", c("Bias Corrected", "Biased"), lty=c(1,3), inset=0.01)
```

---

Coverage

*Sample coverage of a community*

---

**Description**

Calculates an estimator of the sample coverage of a community described by its abundance vector.

**Usage**

```
Coverage(Ns, Estimator = "ZhangHuang", CheckArguments = TRUE)
```

**Arguments**

Ns	A numeric vector containing species abundances.
Estimator	A string containing one of the possible estimators: "ZhangHuang" (default), "Chao", "Turing".
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**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.

**Value**

A number equal to the calculated sample coverage.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Chao, A., Lee, S.-M. and Chen, T.-C. (1988). A generalized Good's nonparametric coverage estimator. *Chinese Journal of Mathematics* 16: 189-199.

Chao, A. and Shen, T.-J. (2010). *Program SPADE: Species Prediction And Diversity Estimation. Program and user's guide*. CARE, Hsin-Chu, Taiwan. <http://chao.stat.nthu.edu.tw/softwareCE.html>

Good, I. J. (1953). On the Population Frequency of Species and the Estimation of Population Parameters. *Biometrika* 40(3/4): 237-264.

Zhang, Z. and Huang, H. (2007). Turing's formula revisited. *Journal of Quantitative Linguistics* 14(2-3): 222-241.

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Ns is the vector of abundances of the metacommunity
Ns <- Paracou618.MC$Ns
# Calculate the sample coverage of the metacommunity
Coverage(Ns) # Stored in Paracou618.SampleCoverage
```

---

Diversity

*Generalized diversity of a community*


---

**Description**

Calculates the generalized diversity of order  $q$  of a probability vector.

**Usage**

```
Diversity(Ps, q = 1, CheckArguments = TRUE)
bcDiversity(Ns, q = 1, Correction = "Best", CheckArguments = TRUE)
```

**Arguments**

Ps	A probability vector, summing to 1.
Ns	A numeric vector containing species abundances.
q	A number: the order of diversity. Default is 1.
Correction	A string containing one of the possible corrections: "None" (no correction), "ChaoShen", "Grassberger", "Holste", "Bonachela", "ZhangGrabchak" or "Best", the default value.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

Diversity calls [Tsallis](#) to calculate entropy and transforms it into diversity by calculating its deformed exponential.

Bias correction requires the number of individuals to estimate sample [Coverage](#). Use `bcDiversity` and choose the `Correction`.

**Value**

A number equal to the calculated diversity.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.

**See Also**

[Tsallis, expq](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Ns is the total number of trees per species
Ns <- Paracou618.MC$Ns
# Whittaker plot
plot(sort(Ns[Ns != 0]), decr = TRUE), log = "y", type = "b",
      main = "Whittaker plot", xlab = "Rank", ylab = "Ns")
# Calculate diversity of order 1, i.e. Shannon's diversity
Diversity(Paracou618.MC$Ps, 1)
# Calculate it with estimation bias correction
bcDiversity(Paracou618.MC$Ns, 1)
```

---

DivEst

*Diversity Estimation of a metacommunity*

---

**Description**

Estimates diversity of a metacommunity.



**Usage**

```
DivEst(q = 0, MC, Biased = TRUE, Correction = "Best", Tree = NULL,
       Normalize = TRUE, Z = NULL, Simulations = 100, CheckArguments = TRUE)
## S3 method for class 'DivEst'
plot(x, ..., main = NULL, Which = "All")
## S3 method for class 'DivEst'
summary(object, ...)
```

**Arguments**

q	A number: the order of diversity.
MC	A <code>MetaCommunity</code> object.
Biased	Logical; if FALSE, a bias correction is applied.
Correction	A string containing one of the possible corrections. The correction must be accepted by <code>DivPart</code> . "Best" is the default value.
Tree	An object of class <code>hclust</code> or <code>phylog</code> . The tree must be ultrametric.
Normalize	If TRUE (default), diversity is not affected by the height of the tree.. If FALSE, diversity is proportional to the height of the tree.
Z	A relatedness matrix, <i>i.e.</i> a square matrix whose terms are all positive, strictly positive on the diagonal. Generally, the matrix is a similarity matrix, <i>i.e.</i> the diagonal terms equal 1 and other terms are between 0 and 1.
Simulations	The number of simulations to build confidence intervals.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
x	An object to be tested or plotted.
main	The title of the plot.
Which	May be "Alpha", "Beta" or "Gamma" to respectively plot the metacommunity's alpha, beta or gamma diversity. If "All" (default), all three plots are shown.
object	A <code>MCdiversity</code> object to be summarized.
...	Additional arguments to be passed to the generic methods.

**Details**

Divest estimates the diversity of the metacommunity and partitions it into alpha and beta components.

If `Tree` is provided, the phylogenetic diversity is calculated else if `Z` is not NULL, then similarity-based entropy is calculated.

Confidence intervals are calculated by Monte-Carlo simulations, drawing simulated communities from a multinomial law following observed frequencies (Marcon et al, 2012 ; 2014)

**Value**

A Divest object which is a [DivPart](#) object with an additional item in its list:

SimulatedDiversity

A matrix containing the simulated values of alpha, beta and gamma diversity.

Divest objects can be summarized and plotted.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf> and Bruno Herault <Bruno.Herault@ecofog.gf>

**References**

Marcon, E., Herault, B., Baraloto, C. and Lang, G. (2012). The Decomposition of Shannon's Entropy and a Confidence Interval for Beta Diversity. *Oikos* 121(4): 516-522.

Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.

Marcon, E., Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).

Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).

**See Also**

[DivPart](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Estimate Shannon diversity.
Estimation <- DivEst(q = 1, Paracou618.MC, Biased = FALSE, Correction = "Best",
  Simulations = 1000)
plot(Estimation)
summary(Estimation)
```

---

DivPart

*Diversity Partition of a metacommunity*

---

**Description**

Partitions the diversity of a metacommunity into alpha and beta components.

**Usage**

```
DivPart(q = 1, MC, Biased = TRUE, Correction = "Best", Tree = NULL,
        Normalize = TRUE, Z = NULL, CheckArguments = TRUE)
## S3 method for class 'DivPart'
plot(x, ...)
## S3 method for class 'DivPart'
summary(object, ...)
```

**Arguments**

q	A number: the order of diversity. Default is 1.
MC	A <a href="#">MetaCommunity</a> object.
Biased	Logical; if FALSE, a bias correction is applied.
Correction	A string containing one of the possible corrections. The correction must be accepted by <a href="#">AlphaEntropy</a> , <a href="#">BetaEntropy</a> and <a href="#">GammaEntropy</a> . "Best" is the default value.
Tree	An object of class <a href="#">hclust</a> or <a href="#">phylog</a> . The tree must be ultrametric.
Normalize	If TRUE (default), diversity is not affected by the height of the tree. If FALSE, diversity is proportional to the height of the tree.
Z	A relatedness matrix, <i>i.e.</i> a square matrix whose terms are all positive, strictly positive on the diagonal. Generally, the matrix is a similarity matrix, <i>i.e.</i> the diagonal terms equal 1 and other terms are between 0 and 1.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
x	An object to be tested or plotted.
object	A <a href="#">MCdiversity</a> object to be summarized.
...	Additional arguments to be passed to the generic methods.

**Details**

DivPart partitions the diversity of the metacommunity into alpha and beta components. It supports estimation-bias correction.

If Tree is provided, the phylogenetic diversity is calculated else if Z is not NULL, then similarity-based entropy is calculated.

Beta diversity/entropy is calculated from Gamma and Alpha when bias correction is required, so community values are not available.

**Value**

A DivPart object. It is a list:

MetaCommunity	The name of the <a href="#">MetaCommunity</a> object containing inventory data.
Order	The value of q.
Biased	Logical. If FALSE, bias corrected values of diversity have been computed.

Correction	The estimation bias correction used to calculate diversity.
Method	The method used to calculate entropy ("HCDT", "Similarity-based").
Tree	The phylogenetic or functional tree used to calculate phylodiversity.
Normalized	Logical. Indicates whether phylodiversity is normalized or proportional to the height of the tree.
Z	The matrix used to calculate similarity-based entropy.
TotalAlphaDiversity	The alpha diversity of communities.
BetaDiversity	The beta diversity of communities.
GammaDiversity	The gamma diversity of the metacommunity.
CommunityAlphaDiversities	A vector containing the alpha diversity of each community.
TotalAlphaEntropy	The alpha entropy of communities.
BetaEntropy	The beta entropy of communities.
GammaEntropy	The gamma entropy of the metacommunity.
CommunityAlphaEntropies	A vector containing the alpha entropy of each community.

DivPart objects can be summarized and plotted.

### Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf> and Bruno Herault <Bruno.Herault@ecofog.gf>

### References

- Marcon, E., Herault, B., Baraloto, C. and Lang, G. (2012). The Decomposition of Shannon's Entropy and a Confidence Interval for Beta Diversity. *Oikos* 121(4): 516-522.
- Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.
- Marcon, E., Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).
- Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).

### See Also

[DivProfile](#)

### Examples

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Estimate Shannon diversity.
summary(DivPart(q = 1, Paracou618.MC, Biased = FALSE) -> dp)
plot(dp)
```

---

 DivProfile

*Diversity Profile of a metacommunity*


---

## Description

Calculate the diversity profiles (alpha, beta, gamma) of a metacommunity.

## Usage

```
DivProfile(q.seq = seq(0, 2, 0.1), MC, Biased = TRUE, Correction = "Best",
  Tree = NULL, Normalize = TRUE, Z = NULL, CheckArguments = TRUE)
## S3 method for class 'DivProfile'
plot(x, ..., main = NULL, xlab = "Order of Diversity",
  ylab = NULL, Which = "All")
## S3 method for class 'DivProfile'
summary(object, ...)
```

## Arguments

q.seq	A numeric vector.
MC	A <a href="#">MetaCommunity</a> object.
Biased	Logical; if FALSE, a bias correction is applied.
Correction	A string containing one of the possible corrections. The correction must be accepted by <a href="#">AlphaEntropy</a> , <a href="#">BetaEntropy</a> and <a href="#">GammaEntropy</a> . "Best" is the default value.
Tree	An object of class <a href="#">hclust</a> or <a href="#">phylog</a> . The tree must be ultrametric.
Normalize	If TRUE (default), diversity is not affected by the height of the tree. If FALSE, diversity is proportional to the height of the tree.
Z	A relatedness matrix, <i>i.e.</i> a square matrix whose terms are all positive, strictly positive on the diagonal. Generally, the matrix is a similarity matrix, <i>i.e.</i> the diagonal terms equal 1 and other terms are between 0 and 1.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
x	An object to be tested or plotted.
main	The main title of the plot. Ignored if Which = "All".
xlab	The x axis label of the plots.
ylab	The y axis label of the plot. Ignored if Which = "All".
Which	May be "Communities", "Alpha", "Beta" or "Gamma" to respectively plot the alpha diversity of communities or the metacommunity's alpha, beta or gamma diversity. If "All" (default), all four plots are shown.
object	A <a href="#">MCdiversity</a> object to be summarized.
...	Additional arguments to be passed to the generic methods.

## Details

If Tree is provided, the phylogenetic diversity is calculated.

DivPart partitions the diversity of the metacommunity into alpha and beta components. It supports estimation-bias correction.

If Tree is provided, the phylogenetic diversity is calculated else if Z is not NULL, then similarity-based entropy is calculated.

Beta diversity/entropy is calculated from Gamma and Alpha when bias correction is required, so community values are not available.

## Value

A DivProfile object. It is a list:

MetaCommunity	The name of the <code>MetaCommunity</code> object containing inventory data.
Order	A vector containing the values of q.
Biased	Logical. If FALSE, bias corrected values of diversity have been computed.
Correction	The estimation bias correction used to calculate diversity.
Method	The method used to calculate entropy ("HCDT", "Similarity-based").
Tree	The phylogenetic or functional tree used to calculate phylodiversity.
Normalized	Logical. Indicates whether phylodiversity is normalized or proportional to the height of the tree.
Z	The matrix used to calculate similarity-based entropy.
CommunityAlphaDiversities	A matrix containing the alpha diversity of each community.
TotalAlphaDiversity	A vector containing the alpha diversity of communities for each order.
BetaDiversity	A vector containing the beta diversity of communities for each order.
GammaDiversity	A vector containing the gamma diversity of the metacommunity for each order.
CommunityAlphaEntropies	A matrix containing the alpha entropy of each community.
TotalAlphaEntropy	A vector containing the alpha entropy of communities for each order.
BetaEntropy	A vector containing the beta entropy of communities for each order.
GammaEntropy	A vector containing the gamma entropy of the metacommunity for each order.

DivProfile objects can be summarized and plotted.

## Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf> and Bruno Herault <Bruno.Herault@ecofog.gf>

## References

Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.

Marcon, E., Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).

Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).

## See Also

[DivPart](#)

## Examples

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Estimate diversity.
Profile <- DivProfile(q.seq = seq(0, 2, 0.1), Paracou618.MC, Biased = FALSE)
plot(Profile)
summary(Profile)
```

---

Dqz

*Similarity-based diversity of a community*

---

## Description

Calculates the diversity of order  $q$  of a probability vector according to a similarity matrix.

## Usage

```
Dqz(Ps, q = 1, Z = diag(length(Ps)), CheckArguments = TRUE)
bcDqz(Ns, q = 1, Z = diag(length(Ns)), Correction = "Best", CheckArguments = TRUE)
```

## Arguments

Ps	A probability vector, summing to 1.
Ns	A numeric vector containing species abundances.
q	A number: the order of diversity. Default is 1.
Z	A relatedness matrix, <i>i.e.</i> a square matrix whose terms are all positive, strictly positive on the diagonal. Generally, the matrix is a similarity matrix, <i>i.e.</i> the diagonal terms equal 1 and other terms are between 0 and 1. Default is the identity matrix to calculate neutral diversity.
Correction	A string containing one of the possible corrections: "None" (no correction), "HorvitzThomson", "MarconZhang" or "Best", the default value. The "MarconZhang" correction assumes a similarity matrix.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

Diversity is calculated following Leinster and Cobbold (2012): it is the reciprocal of the (generalized) average (of order  $q$ ) of the community species ordinariness.

A similarity matrix is used (as for [Dqz](#)), not a distance matrix as in Ricotta and Szeidl (2006). See the example.

Bias correction requires the number of individuals. Use `bcHqz` and choose the `Correction`. Correction techniques are from Marcon and Zhang (2014).

Currently, the "Best" correction is the max value of "HorvitzThomson" and "MarconZhang".

**Value**

A number equal to the calculated diversity.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Leinster, T. and Cobbold, C. (2012). Measuring diversity: the importance of species similarity. *Ecology* 93(3): 477-489.

Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).

**See Also**

[Hqz, PhyloDiversity](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Prepare the similarity matrix
DistanceMatrix <- as.matrix(Paracou618.dist)
# Similarity can be 1 minus normalized distances between species
Z <- 1 - DistanceMatrix/max(DistanceMatrix)
# Calculate diversity of order 2
bcDqz(Paracou618.MC$Ns, 2, Z)
```



---

EightSpAbundance	<i>Abundances of 8 species to run examples.</i>
------------------	-------------------------------------------------

---

**Description**

This dataset is a light-weight example.

**Usage**

```
data(Paracou618)
```

**Format**

A named vector.

**Examples**

```
data(Paracou618)
EightSpAbundance
```

---

EightSpTree	<i>Functional tree with 8 species.</i>
-------------	----------------------------------------

---

**Description**

This dataset is a light-weight example.

**Usage**

```
data(Paracou618)
```

**Format**

An object of class `phylog` containing a functional tree.

**Examples**

```
data(Paracou618)
plot(EightSpTree)
```

---

Enq

*Grassberger's expectation of  $n^q$*

---

**Description**

Expected value of  $n^q$  when  $n$  follows a Poisson law.

**Usage**

Enq(n, q)

**Arguments**

n	A positive integer vector.
q	A positive number.

**Details**

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

**Value**

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

**Note**

The function is computed using the [beta](#) function.

Its value is 0 for  $n - q + 1 < 0$ .

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Grassberger, P. (1988). Finite sample corrections to entropy and dimension estimates. *Physics Letters A* 128(6-7): 369-373.

EntropyCI

*Entropy of Monte-Carlo simulated communities***Description**

Resamples a community by Monte-Carlo simulations of a multinomial distribution and returns a vector of entropy values to calculate confidence intervals.

**Usage**

```
EntropyCI(FUN, Simulations = 100, Ns, ..., CheckArguments = TRUE)
```

**Arguments**

FUN	The entropy function to be applied to each simulated community. May be any entropy function accepting a vector of species abundances, such as <a href="#">bcTsallis</a> , <a href="#">bcShannon</a> , <a href="#">bcSimpson</a> or <a href="#">bcPhyloEntropy</a> .
Simulations	The number of simulations to build confidence intervals.
Ns	A numeric vector containing species abundances.
...	Additional arguments to be passed to FUN.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

This function is used to obtain the distribution of entropy and eventually calculate confidence intervals. It draws simulated communities according to a multinomial distribution with the same number of individuals and probabilities as the actual community. It calculates the entropy of each simulated community. Last, it recenters the distribution of entropy values around the actual value of entropy according to Marcon et al. (2012): the estimation bias of simulated communities entropy can not be corrected analytically, but it does not affect the distribution shape.

Diversity can not be recentered this way so diversity function should not be used. Unexpected results will be obtained if inappropriate functions are used.

**Value**

A numeric vector containing the entropy value of each simulated community.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Marcon, E., Herault, B., Baraloto, C. and Lang, G. (2012). The Decomposition of Shannon's Entropy and a Confidence Interval for Beta Diversity. *Oikos* 121(4): 516-522.

**Examples**

```

# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Abundance
Ns <- Paracou618.MC$Ns
q <- 1
# Estimate entropy and transform it into diversity
RealEst <- expq(bcTsallis(Ns, q), q)
# Transform the distribution of Tsallis entropy into diversity
SimulatedDiversity <- expq(EntropyCI(bcTsallis, Simulations=1000, Ns, q), q)
# Figure
plot(density(SimulatedDiversity), col="black", lwd=2, main="", xlab="Diversity")
abline(v=RealEst, col="red", lwd=2, lty=2)
abline(v=quantile(SimulatedDiversity, probs = 0.025), col="black", lwd=1, lty=3)
abline(v=quantile(SimulatedDiversity, probs = 0.975), col="black", lwd=1, lty=3)
legend("topright", c("Real value", "Confidence interval"), lty=c(2,3),
      col=c("red", "black"), inset=0.01)
# Print results
cat("Estimated Diversity:", RealEst)
quantile(SimulatedDiversity, probs = c(0.025, 0.975))

```

expq

*Exponential of order q***Description**

Calculates the deformed exponential of order  $q$ .

**Usage**

```
expq(x, q)
```

**Arguments**

x	A numeric vector.
q	A number.

**Details**

The deformed exponential 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.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

## References

- Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.
- Tsallis, C. (1994). What are the numbers that experiments provide? *Quimica Nova* 17(6): 468-471.

## See Also

[expq](#)

## Examples

```
curve(exp(x), -5, 0, lty=3)
curve(expq(x, 2), -5, 0, lty=2, add=TRUE)
curve(expq(x, 3), -5, 0, lty=1, add=TRUE)
legend("topleft", legend = c("exp(x)", "exp2(x)", "exp3(x)"), lty = c(1, 2, 3), inset=0.02)
```

---

GammaDiversity	<i>Unbiased gamma diversity of a metacommunity</i>
----------------	----------------------------------------------------

---

## Description

Calculates the unbiased diversity of order  $q$  of a metacommunity.

## Usage

```
GammaDiversity(MC, q = 1, Correction = "Best", Tree = NULL, Normalize = TRUE,
  Z = NULL, CheckArguments = TRUE)
```

## Arguments

MC	A <a href="#">MetaCommunity</a> object.
q	A number: the order of diversity. Default is 1.
Correction	A string containing one of the possible corrections: "ChaoShen", "Grassberger", "Holste", "Bonache1a" or "Best", the default value.
Tree	An object of class <a href="#">hclust</a> or <a href="#">phylog</a> . The tree must be ultrametric.
Normalize	If TRUE (default), diversity is not affected by the height of the tree. If FALSE, diversity is proportional to the height of the tree.
Z	A relatedness matrix, <i>i.e.</i> a square matrix whose terms are all positive, strictly positive on the diagonal. Generally, the matrix is a similarity matrix, <i>i.e.</i> the diagonal terms equal 1 and other terms are between 0 and 1.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

## Details

Entropy is calculated by [GammaEntropy](#) and transformed into diversity.

**Value**

The metacommunity's gamma entropy.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.

Marcon, E., Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).

Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).

**See Also**

[GammaEntropy](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Calculate Simpson gamma diversity
GammaDiversity(Paracou618.MC, 2)
# Compare without correction
GammaDiversity(Paracou618.MC, 2, Correction = "None")
# Estimate phylogenetic Simpson gamma diversity
GammaDiversity(Paracou618.MC, 2, Tree = Paracou618.Taxonomy)
```

---

GammaEntropy

*Unbiased gamma entropy of a metacommunity*

---

**Description**

Calculates the unbiased Tsallis entropy of order  $q$  of a metacommunity.

**Usage**

```
GammaEntropy(MC, q = 1, Correction = "Best", Tree = NULL, Normalize = TRUE,
  Z = NULL, CheckArguments = TRUE)
```

**Arguments**

MC	A <a href="#">MetaCommunity</a> object.
q	A number: the order of entropy. Default is 1.
Correction	A string containing one of the possible corrections accepted by the bias-corrected entropy function (see details) or "None" or "Best", the default value.
Tree	An object of class <a href="#">hclust</a> or <a href="#">phylog</a> . The tree must be ultrametric.
Normalize	If TRUE (default), the entropy returned by the function is normalized by the height of the tree (it is the weighted average value of the entropy in each slice). If FALSE, it is the unnormalized weighted sum of the results.
Z	A relatedness matrix, <i>i.e.</i> a square matrix whose terms are all positive, strictly positive on the diagonal. Generally, the matrix is a similarity matrix, <i>i.e.</i> the diagonal terms equal 1 and other terms are between 0 and 1.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

If Tree is not NULL, then phylogenetic entropy is calculated by [bcPhyloEntropy](#) else if Z is not NULL, then similarity-based entropy is calculated by [bcHqz](#), else neutral entropy is calculated by [bcTsallis](#), else .

**Value**

A number equal to the calculated entropy.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.

Marcon, E., Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).

Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).

**See Also**

[bcTsallis](#), [bcPhyloEntropy](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Calculate Simpson gamma entropy
GammaEntropy(Paracou618.MC, 2)
# Compare without correction
GammaEntropy(Paracou618.MC, 2, Correction = "None")
# Estimate phylogenetic Simpson gamma entropy
GammaEntropy(Paracou618.MC, 2, Tree = Paracou618.Taxonomy)
```

Hqz

*Similarity-based entropy of a community***Description**

Calculates the entropy of order  $q$  of a probability vector according to a similarity matrix.

**Usage**

```
Hqz(Ps, q = 1, Z = diag(length(Ps)), CheckArguments = TRUE)
bcHqz(Ns, q = 1, Z = diag(length(Ns)), Correction = "Best", CheckArguments = TRUE)
```

**Arguments**

Ps	A probability vector, summing to 1.
Ns	A numeric vector containing species abundances.
q	A number: the order of entropy. Default is 1.
Z	A relatedness matrix, <i>i.e.</i> a square matrix whose terms are all positive, strictly positive on the diagonal. Generally, the matrix is a similarity matrix, <i>i.e.</i> the diagonal terms equal 1 and other terms are between 0 and 1. Default is the identity matrix to calculate neutral entropy.
Correction	A string containing one of the possible corrections: "None" (no correction), "ChaoShen", "MarconZhang" or "Best", the default value. The "MarconZhang" correction assumes a similarity matrix.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

Entropy is calculated following Leinster and Cobbold (2012) after Ricotta and Szeidl (2006): it is the entropy of order  $q$  of the community, using species ordinariness as the information function.

A similarity matrix is used (as for [Dqz](#)), not a distance matrix as in Ricotta and Szeidl (2006). See the example.

Bias correction requires the number of individuals. Use `bcHqz` and choose the `Correction`. Correction techniques are from Marcon and Zhang (2014).

Currently, the "Best" correction is the max value of "ChaoShen" and "MarconZhang".



**Value**

A number equal to the calculated entropy.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Leinster, T. and Cobbold, C. (2012). Measuring diversity: the importance of species similarity. *Ecology* 93(3): 477-489.

Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).

Ricotta, C. and Szeidl, L. (2006). Towards a unifying approach to diversity measures: Bridging the gap between the Shannon entropy and Rao's quadratic index. *Theoretical Population Biology* 70(3): 237-243.

**See Also**

[Dqz](#), [PhyloEntropy](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Prepare the similarity matrix
DistanceMatrix <- as.matrix(EightSpTree$Wdist^2/2)
# Similarity can be 1 minus normalized distances between species
Z <- 1 - DistanceMatrix/max(DistanceMatrix)
# Calculate diversity of order 2
Ps <- EightSpAbundance/sum(EightSpAbundance)
Hqz(Ps, 2, Z)
# Equal to normalized Rao quadratic entropy when q=2
Rao(Ps, EightSpTree)/max(DistanceMatrix)
# But different from PhyloEntropy for all other q, e.g. 1
Hqz(Ps, 1, Z)
summary(PhyloEntropy(Ps, 1, EightSpTree))
```

---

HqzBeta

*Similarity-based beta entropy of a community*

---

**Description**

Calculates the similarity-based beta entropy of order  $q$  of a community belonging to a metacommunity.

**Usage**

```
HqzBeta(Ps, Pexp = NULL, q = 1, Z = diag(length(Ps)), CheckArguments = TRUE)
bcHqzBeta(Ns, Nexp = NULL, q = 1, Z = diag(length(Ns)), Correction = "Best",
          CheckArguments = TRUE)
```

**Arguments**

Ps	The probability vector of species of the community.
Pexp	The probability vector of species of the metacommunity.
Ns	A numeric vector containing species abundances of the community.
Nexp	A numeric vector containing species abundances of the metacommunity.
q	A number, the order of diversity. Default is 1.
Z	A relatedness matrix, <i>i.e.</i> a square matrix whose terms are all positive, strictly positive on the diagonal. Generally, the matrix is a similarity matrix, <i>i.e.</i> the diagonal terms equal 1 and other terms are between 0 and 1. Default is the identity matrix to calculate neutral entropy.
Correction	A string containing one of the possible corrections: currently, no correction is available so "Best", the default value, is equivalent to "None".
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

The derivation of similarity-based beta entropy can be found in Marcon and Zhang (2014).

Bias correction requires the number of individuals.

Note that beta entropy value is related to alpha entropy (if  $q$  is not 1) and cannot be compared across communities (Jost, 2007). Beta entropy of a community is not meaningful in general, do rather calculate the [BetaDiversity](#) of the metacommunity.

**Value**

A number equal to the calculated entropy.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Jost (2007), Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.

Marcon, E. and Zhang, Z. (2014). The decomposition of similarity-based diversity and its bias correction. *HAL* hal-00989454(version 1).

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Probability distribution of the first plot
Ps1 <- Paracou618.MC$Psi[, 1]
# Prepare the similarity matrix
DistanceMatrix <- as.matrix(Paracou618.dist)
# Similarity can be 1 minus normalized distances between species
Z <- 1 - DistanceMatrix/max(DistanceMatrix)
# Divergence of order 2 between plot 1 and the whole forest
HqzBeta(Ps1, Ps, 2, Z)
```

KLq

*Generalized Kullback-Leibler divergence***Description**

Calculates the generalized Kullback-Leibler divergence between an observed and an expected probability distribution.

**Usage**

```
KLq(Ps, Pexp, q = 1, CheckArguments = TRUE)
```

**Arguments**

Ps	The observed probability vector.
Pexp	The expected probability vector.
q	A number: the order of entropy. Default is 1.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

The generalized Kullback-Leibler divergence (Borland et al., 1998) converges to the Kullback-Leibler divergence (Kullback and Leibler, 1951) when  $q$  tends to 1. It is used to calculate the generalized beta entropy (Marcon et al., submitted).

**Value**

A number equal to the generalized Kullback-Leibler divergence between the probability distributions.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

## References

- Borland, L., Plastino, A. R. and Tsallis, C. (1998). Information gain within nonextensive thermostatics. *Journal of Mathematical Physics* 39(12): 6490-6501.
- Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics* 22(1): 79-86.
- Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.

## See Also

[TsallisBeta](#)

## Examples

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Probability distribution of the first plot
Ps1 <- Paracou618.MC$Psi[, 1]
# Divergence of order 2 between the first plot and the whole forest
KLq(Ps1, Ps, 2)
```

---

ln<sub>q</sub>

*Logarithm of order q*

---

## Description

Calculates the deformed logarithm of order  $q$ .

## Usage

ln<sub>q</sub>(x, q)

## Arguments

x	A numeric vector.
q	A number.

## Details

The deformed logarithm 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.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Tsallis, C. (1994). What are the numbers that experiments provide? *Quimica Nova* 17(6): 468-471.

**See Also**

[expq](#)

**Examples**

```
curve(log(x), 0, 1, lty=1)
curve(lnq(x, 2), 0, 1, lty=2, add=TRUE)
curve(lnq(x, 3), 0, 1, lty=3, add=TRUE)
legend("topleft", legend = c("log(x)", "ln2(x)", "ln3(x)"), lty = c(1, 2, 3), inset=0.02)
```

---

MC Utilities

*Manipulation of meta-communities*

---

**Description**

Tools to manipulate meta-communities. From a list of meta-communities, MergeMC creates a meta-community whose communities are each original metacommunity. MergeC creates a metacommunity whose communities are each original community. ShuffleMC randomly assigns original communities to a metacommunity, keeping original weights, and returns a list of meta-communities.

**Usage**

```
MergeMC(MClist, Weights = rep(1, length(MClist)), CheckArguments = TRUE)
MergeC(MClist, Weights = rep(1, length(MClist)), CheckArguments = TRUE)
ShuffleMC(MClist, Weights = rep(1, length(MClist)), CheckArguments = TRUE)
```

**Arguments**

MClist	A list of MetaCommunity objects.
Weights	A vector of numbers containing the weight of each metacommunity of the list. It does not have to be normalized to sum to 1.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

MergeMC is used for hierarchical partitioning of diversity. The gamma diversity of communities of the list becomes alpha diversity of the merged meta-community.

MergeC creates a new meta-community by mixing original ones. Original communities are kept, their new weight is the product of their original weight and the weight of their original meta-community.

ShuffleMC is used for simulations of the null hypothesis that all metacommunities of the list are identical.

**Value**

MergeMC and MergeC return a [MetaCommunity](#).

ShuffleMC returns a [list](#) of [MetaCommunity](#) objects.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**See Also**

[MetaCommunity](#)

**Examples**

```
# First meta-community
(df <- data.frame(C1 = c(10, 10, 10, 10), C2 = c(0, 20, 35, 5),
  C3 = c(25, 15, 0, 2), row.names = c("sp1", "sp2", "sp3", "sp4")))
w <- c(1, 2, 1)
MC1 <- MetaCommunity(Abundances = df, Weights = w)
# Second meta-community
(df <- data.frame(C1 = c(10, 4), C2 = c(3, 4), row.names = c("sp1", "sp5")))
w <- c(3, 2)
MC2 <- MetaCommunity(Abundances = df, Weights = w)

# Merge communities
plot(MergeC(list(MC1, MC2)), main="Merged communities")
# Merge metacommunities
plot(MergeMC(list(MC1, MC2)), main="Merged meta-communities")
```

---

MCdiversity

*Meta-Community diversity class.*

---

**Description**

Methods for objects of type "MCdiversity".

**Usage**

```
is.MCdiversity(x)
## S3 method for class 'MCdiversity'
plot(x, ...)
## S3 method for class 'MCdiversity'
summary(object, ...)
```

**Arguments**

x	An object to be tested or plotted.
object	A MCdiversity object to be summarized.
...	Additional arguments to be passed to the generic methods.

**Value**

Meta-community diversity objects are lists containing:

MetaCommunity	The name of the <a href="#">MetaCommunity</a> object containing inventory data.
Type	The type of diversity ("alpha", "beta" or "gamma").
Order	The order of diversity $q$ .
Correction	The estimation bias correction used to calculate diversity.
Tree	The phylogenetic or functional tree used to calculate phylodiversity.
Normalized	Logical. Indicates whether phylodiversity is normalized or proportional to the height of the tree.
Weights	A vector containing the weights of communities.
Communities	A vector containing the diversity of communities.
Total	The total diversity.

`is.MCdiversity` returns TRUE if the object is of class `MCdiversity`.

`summary.MCdiversity` returns a summary of the object's value.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

---

MEntropy

*Meta-Community entropy class.*


---

### Description

Methods for objects of type "MEntropy".

### Usage

```
is.MEntropy(x)
## S3 method for class 'MEntropy'
plot(x, ...)
## S3 method for class 'MEntropy'
summary(object, ...)
```

### Arguments

x	An object to be tested or plotted.
object	A MEntropy object to be summarized.
...	Additional arguments to be passed to the generic methods.

### Value

Meta-community entropy objects are lists containing:

MetaCommunity	The name of the <a href="#">MetaCommunity</a> object containing inventory data.
Method	The method used to calculate entropy ("HCDT", "Similarity-based").
Type	The type of entropy ("alpha", "beta" or "gamma").
Order	The order of entropy $q$ .
Correction	The estimation bias correction used to calculate entropy.
Tree	The phylogenetic or functional tree used to calculate phyloentropy.
Normalized	Logical. Indicates whether phyloentropy is normalized or proportional to the height of the tree.
Z	The matrix used to calculate similarity-based entropy.
Weights	A vector containing the weights of communities.
Communities	A vector containing the entropy of communities.
Total	The total entropy.

is.MEntropy returns TRUE if the object is of class MEntropy.

summary.MEntropy returns a summary of the object's value.

### Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf>



---

MetaCommunity	<i>Metacommunity class</i>
---------------	----------------------------

---

## Description

Methods for objects of type "MetaCommunity".

## Usage

```
MetaCommunity(Abundances, Weights = rep(1, ncol(Abundances)))
is.MetaCommunity(x)
## S3 method for class 'MetaCommunity'
summary(object, ...)
## S3 method for class 'MetaCommunity'
plot(x, ...)
```

## Arguments

Abundances	A dataframe containing the number of observations (lines are species, columns are communities). The first column of the dataframe may contain the species names.
Weights	A vector of positive numbers equal to community weights or a dataframe containing a vector named Weights. It does not have to be normalized. Weights are equal by default.
x	An object to be tested or plotted.
object	A MetaCommunity object to be summarized.
...	Additional arguments to be passed to the generic methods.

## Details

In the entropart package, individuals of different "species" are counted in several "communities" which are aggregated to define a "metacommunity".

This is a naming convention, which may correspond to plots in a forest inventory or any data organized the same way.

Alpha and beta entropies of communities are summed according to Weights and the probability to find a species in the metacommunity is the weighted average of probabilities in communities.

The simplest way to import data is to organize it into two text files. The first file should contain abundance data: the first column named Species for species names, and a column for each community.

Species	NameOfCommunity1	NameOfCommunity2
NameOfSpecies1	1	5
NameOfSpecies2	4	2
...	...	...

The second file should contain the community weights and be organized as follows:

Communities	Weights
NameOfCommunity1	3
NameOfCommunity2	1

Files can be read and data imported by code such as:

```
Abundances <- read.csv(file="Abundances.csv")
Weights <- read.csv(file="Weights.csv")
MC <- MetaCommunity(Abundances, Weights)
```

## Value

An object of class **MetaCommunity** is a list:

Nsi	A matrix containing abundance data, species in line, communities in column.
Ns	A vector containing the number of individuals of each species.
Ni	A vector containing the number of individuals of each community.
N	The total number of individuals.
Psi	A matrix whose columns are the probability vectors of communities (each of them sums to 1).
Wi	A vector containing the normalized community weights (sum to 1).
Ps	A vector containing the probability vector of the metacommunity.
Nspecies	The number of species.
Ncommunities	The number of communities.
SampleCoverage	The sample coverage of the metacommunity.
SampleCoverage.communities	A vector containing the sample coverages of each community.

`is.MetaCommunity` returns TRUE if the object is of class `MetaCommunity`.

`summary.MetaCommunity` returns a summary of the object's value.

`plot.MetaCommunity` plots it.

## Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf>

## Examples

```
# Use BCI data from vegan package
if (require(vegan, quietly = TRUE)) {
  # Load BCI data (number of trees per species in each 1-ha plot of a tropical forest)
  data(BCI)
  # BCI dataframe must be transposed (its lines are plots, not species)
```

```
BCI.df <- as.data.frame(t(BCI))
# Create a metacommunity object from a matrix of abundances and a vector of weights
# (here, all plots have a weight equal to 1)
MC <- MetaCommunity(BCI.df, rep(1,50))
}
```

---

Originality.Species    *Originality of species*

---

### Description

Calculates the originality of species according to the order of diversity.

### Usage

```
Originality.Species(Tree, q = 2, CheckArguments = TRUE)
```

### Arguments

Tree	An object of class <a href="#">hclust</a> or <a href="#">phylog</a> . The tree must be ultrametric.
q	A number: the order of diversity. Default is 2 for Rao quadratic entropy.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

### Details

The originality of species is their frequency resulting in the maximum value of generalized entropy of order  $q$ . Originality was introduced by Pavoine et al. (2005) for Rao quadratic entropy ( $q = 2$ ). It can be calculated exactly by [originality](#). For other orders of diversity, numeric optimization is required.

### Value

A named vector of frequencies.

### Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf>

### References

Pavoine, S., Ollier, S. and Dufour, A.-B. (2005). Is the originality of a species measurable? *Ecology Letters* 8: 579-586.

### See Also

[originality](#)

**Examples**

```

data(Paracou618)
# Calculate Originality of a species according to a simple functional tree, q=1
(Frequencies <- Originality.Species(EightSpTree, 1))
# Plot the result
if (require(ade4, quietly = TRUE)) {
  dotchart.phylog(EightSpTree, Frequencies)
}

```

---

Paracou618.dist	<i>Functional distances between pairs of species of Paracou field station plots 6 and 18, two 1-ha plots inventoried by the Bridge project.</i>
-----------------	-------------------------------------------------------------------------------------------------------------------------------------------------

---

**Description**

This dataset is from Paracou field station, French Guiana, managed by **Cirad**. Traits are detailed in Marcon and Herault (2014), the distance matrix was built following Paine et al. (2011).

**Usage**

```
data(Paracou618)
```

**Format**

An object of class `dist`.

**Source**

Permanent data census of Paracou.

**References**

Gourlet-Fleury, S., Guehl, J. M. and Laroussinie, O., Eds. (2004). *Ecology & management of a neotropical rainforest. Lessons drawn from Paracou, a long-term experimental research site in French Guiana*. Paris, Elsevier.

Marcon, E. and Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).

Paine, C. E. T., Baraloto, C., Chave, J., and Herault, B. (2011). Functional traits of individual trees reveal ecological constraints on community assembly in tropical rain forests. *Oikos*, 120(5), 720-727.

**Examples**

```

data(Paracou618)
plot(density(Paracou618.dist, from=0), main="Distances between species")

```

---

Paracou618.Functional *Functional tree of species of Paracou field station plots 6 and 18, two 1-ha plots inventoried by the Bridge project.*

---

## Description

This dataset is from Paracou field station, French Guiana, managed by [Cirad](#). Traits are detailed in Marcon and Herault (2014), the tree was built following Paine et al. (2011), based on [Paracou618.dist](#).

## Usage

```
data(Paracou618)
```

## Format

An object of class [hclust](#).

## Source

Permanent data census of Paracou.

## References

Gourlet-Fleury, S., Guehl, J. M. and Laroussinie, O., Eds. (2004). *Ecology & management of a neotropical rainforest. Lessons drawn from Paracou, a long-term experimental research site in French Guiana*. Paris, Elsevier.

Marcon, E. and Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).

Paine, C. E. T., Baraloto, C., Chave, J., and Herault, B. (2011). Functional traits of individual trees reveal ecological constraints on community assembly in tropical rain forests. *Oikos*, 120(5), 720-727.

## Examples

```
data(Paracou618)  
plot(Paracou618.Functional)
```

---

Paracou618.MC	<i>Paracou field station plots 6 and 18, two 1-ha plots inventoried by the Bridge project.</i>
---------------	------------------------------------------------------------------------------------------------

---

**Description**

This dataset is from Paracou field station, French Guiana, managed by [Cirad](#).

**Usage**

```
data(Paracou618)
```

**Format**

An object of class [MetaCommunity](#) made of two communities and 425 species.

**Source**

Permanent data census of Paracou and Marcon et al. (2012).

**References**

Gourlet-Fleury, S., Guehl, J. M. and Laroussinie, O., Eds. (2004). *Ecology & management of a neotropical rainforest. Lessons drawn from Paracou, a long-term experimental research site in French Guiana*. Paris, Elsevier.

Marcon, E., F. Puech, et al. (2012). Characterizing the relative spatial structure of point patterns. *International Journal of Ecology* 2012(Article ID 619281): 11.

**Examples**

```
data(Paracou618)
Paracou618.MC
```

---

Paracou618.Taxonomy	<i>Taxonomy (Family - Genus - Species) of Paracou field station plots 6 and 18, two 1-ha plots inventoried by the Bridge project.</i>
---------------------	---------------------------------------------------------------------------------------------------------------------------------------

---

**Description**

This dataset is from Paracou field station, French Guiana, managed by [Cirad](#).

**Usage**

```
data(Paracou618)
```

**Format**

An object of class `phylog` containing a taxonomy.

**Source**

Permanent data census of Paracou.

**References**

Gourlet-Fleury, S., Guehl, J. M. and Laroussinie, O., Eds. (2004). *Ecology & management of a neotropical rainforest. Lessons drawn from Paracou, a long-term experimental research site in French Guiana*. Paris, Elsevier.

**Examples**

```
data(Paracou618)
plot(Paracou618.Taxonomy)
```

---

PDFD

*Phylogenetic Diversity / Functional Diversity of a Community*

---

**Description**

Calculates Faith's PD / Petchey and Gaston' FD of a community described by a probability vector and a phylogenetic / functional tree.

**Usage**

```
PDFD(Ps, Tree, CheckArguments = TRUE)
```

**Arguments**

Ps	A probability vector, summing to 1.
Tree	An object of class <code>hclust</code> or <code>phylog</code> . The tree must be ultrametric.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

PD and FD are defined as the total length of the branches of the tree.

The probability vector is used to select branches: branches with probability 0 are eliminated.

Bias correction requires the number of individuals to estimate sample [Coverage](#).

Use `bcPhyloDiversity(Ps, 0, Tree)` and choose the Correction.

**Value**

A number equal to the calculated diversity.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Faith, D. P. (1992). Conservation evaluation and phylogenetic diversity. *Biological Conservation* 61(1): 1-10.

Petchey, O. L. and Gaston, K. J. (2002). Functional diversity (FD), species richness and community composition. *Ecology Letters* 5: 402-411.

**See Also**

[bcPhyloDiversity](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest
# and their taxonomy)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Calculate the phylogenetic Shannon diversity of the plot
PDFD(Ps, Paracou618.Taxonomy)
```

---

PhyloApply

*Apply a Function over a Phylogenetic Tree*

---

**Description**

Cuts the tree into slices separated by nodes, applies the function to each slice and returns the weighted (by slice lengths) sum of the results.

**Usage**

```
PhyloApply(Tree, FUN, NorP, Normalize = TRUE, ..., CheckArguments = TRUE)
```

**Arguments**

Tree	An object of class <a href="#">hclust</a> or <a href="#">phylog</a> or PPTree. The tree must be ultrametric.
FUN	The function to be applied to each interval of the tree.
NorP	A numeric vector or a two-column matrix. Contains either abundances or probabilities. Two-column matrices should contain the observed abundances (or probabilities) in the first column and the expected ones in the second column, to allow using beta diversity functions.



Normalize	If TRUE (default), the Total value returned by Function is normalized by the height of the tree (it is the weighted average value of the result in each slice). If FALSE, it is the unnormalized weighted sum of the results.
...	Further arguments to pass to Function.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

### Details

This function is generally not used directly. It is a tool to calculate [PhyloEntropy](#) and [PhyloDiversity](#).

Intervals separate two cuts in a tree: no node is found at heights contained in an interval.

Objects of class **PPtree** are returned by [Preprocess.Tree](#).

### Value

An object of class PhyloValue. It is a list:

Distribution	The distribution used to calculate the value
Function	The function used to calculate the value
Tree	The functional or phylogenetic tree used to calculate the value
Normalized	Logical. Indicates whether phylovalue is normalized or proportional to the height of the tree.
Cuts	A named vector containing values along the tree. Names are cut ends, <i>i.e.</i> the ends of intervals (the first interval starts at 0 for leaves, the max value is the height of the tree).
Total	A value equal the total value, multiplied by the tree height if Normalize is FALSE.

### Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf>

### References

Marcon, E., Herault, B. (2014). Decomposing PhyloDiversity. *HAL* hal-00946177(version 1).

### See Also

[Preprocess.Tree](#)

### Examples

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest
# and their taxonomy)
data(Paracou618)
# Plot the taxonomy
if (require(ade4, quietly = TRUE)) {
  radial.phylog(Paracou618.Taxonomy)
```

```

}
# Calculate the mean number of trees (individuals) per species
# (Cuts are 1=species, 2=genus, 3=family)
PhyloApply(Paracou618.Taxonomy, mean, Paracou618.MC$Ns, TRUE)

```

---

PhyloBetaEntropy      *Phylogenetic Beta Entropy of a community*

---

## Description

Calculates the phylogenetic beta entropy of order  $q$  of a community belonging to a metacommunity.

## Usage

```

PhyloBetaEntropy(Ps, Pexp, q = 1, Tree, Normalize = TRUE, CheckArguments = TRUE)
bcPhyloBetaEntropy(Ns, Nexp, q = 1, Tree, Normalize = TRUE, Correction = "Best",
  CheckArguments = TRUE)

```

## Arguments

Ps	The probability vector of species of the community.
Pexp	The probability vector of species of the metacommunity.
Ns	A numeric vector containing species abundances of the community.
Nexp	A numeric vector containing species abundances of the metacommunity.
q	A number: the order of entropy. Default is 1.
Tree	An object of class <code>hclust</code> or <code>phylog</code> . The tree must be ultrametric.
Normalize	If TRUE (default), the entropy returned by the function is normalized by the height of the tree (it is the weighted average value of the entropy in each slice). If FALSE, it is the unnormalized weighted sum of the results.
Correction	A string containing one of the possible corrections: currently, only "ChaoShen". "Best" is the default value, it is equivalent to "ChaoShen".
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

## Details

The phylogenetic entropy is the generalization of HCDT entropy to unequal species distances (Pavoine et al., 2009).

Calculation relies on [TsallisBeta](#) and [PhyloApply](#).

Bias correction requires the number of individuals to estimate sample [Coverage](#). Use `bcPhyloBetaEntropy` and choose the `Correction`.

Note that beta entropy value is related to alpha entropy (if  $q$  is not 1) and cannot be compared across communities (Jost, 2007). Beta entropy of a community is not meaningful in general, do rather calculate the [PhyloDiversity](#) of the metacommunity.

**Value**

A [PhyloEntropy](#) object containing entropy values at each cut of the tree.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Jost (2007), Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.

Marcon, E., Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).

Pavoine, S., Love, M. S. and Bonsall, M. B. (2009). Hierarchical partitioning of evolutionary and ecological patterns in the organization of phylogenetically-structured species assemblages: Application to rockfish (genus: *Sebastes*) in the Southern California Bight. *Ecology Letters* 12(9): 898-908.

**See Also**

[TsallisBeta](#), [bcPhyloBetaEntropy](#), [PhyloDiversity](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest
# and their taxonomy)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Probability distribution of the first plot
Ps1 <- Paracou618.MC$Psi[, 1]
# Calculate the phylogenetic Shannon beta entropy of the plot
summary(PhyloBetaEntropy(Ps1, Ps, 1, Paracou618.Taxonomy) -> e)
plot(e)

# Ns is the vector of abundances of the metacommunity
Ns <- Paracou618.MC$Ns
# Abundances in the first plot
Ns1 <- Paracou618.MC$Nsi[, 1]
# Calculate the phylogenetic Shannon beta entropy of the plot
summary(bcPhyloBetaEntropy(Ns1, Ns, 1, Paracou618.Taxonomy, Correction = "Best") -> e)
plot(e)
```

---

PhyloDiversity      *Phylogenetic Diversity of a Community*

---

### Description

Calculates the phylogenetic diversity of order  $q$  of a probability vector.

### Usage

```
PhyloDiversity(Ps, q = 1, Tree, Normalize = TRUE, CheckArguments = TRUE)
bcPhyloDiversity(Ns, q = 1, Tree, Normalize = TRUE, Correction = "Best",
  CheckArguments = TRUE)
is.PhyloDiversity(x)
## S3 method for class 'PhyloDiversity'
summary(object, ...)
```

### Arguments

Ps	A probability vector, summing to 1.
Ns	A numeric vector containing species abundances.
q	A number: the order of diversity. Default is 1.
Tree	An object of class <code>hclust</code> or <code>phylog</code> . The tree must be ultrametric.
Normalize	If TRUE (default), the Total diversity is not affected by the height of the tree. If FALSE, it is proportional to the height of the tree.
Correction	A string containing one of the possible corrections: "None" (no correction), "ChaoShen", "Grassberger", "Holste", "Bonachela" or "Best", the default value.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
x	An object to be tested or plotted
object	A PhyloDiversity object to be summarized.
...	Additional arguments to be passed to the generic methods.

### Details

The phylogenetic entropy is its generalization of HCDT entropy to unequal species distances (Pavoine et al., 2009).

Diversity is obtained by transforming generalized entropy.

Bias correction requires the number of individuals to estimate sample [Coverage](#). Use `bcPhyloDiversity` and choose the `Correction`.

**Value**

An object of class PhyloDiversity is a list:

Distribution	The distribution used to calculate diversity
Function	The function used to calculate diversity
Tree	The functional or phylogenetic tree used to calculate diversity
Normalized	Logical. Indicates whether phylodiversity is normalized or proportional to the height of the tree.
Type	The type of diversity ("alpha", "beta" or "gamma").
Order	The order of diversity $q$ .
Cuts	A named vector containing values of neutral diversity along the tree. Names are cut ends, <i>i.e.</i> the ends of intervals (the first interval starts at 0 for leaves, the max value is the height of the tree).
Total	A value equal the total diversity (obtained by transforming the total normalized entropy), multiplied by the tree height if <code>Normalize</code> is FALSE.

`is.PhyloDiversity` returns TRUE if the object is of class PhyloDiversity.

`summary.PhyloDiversity` returns a summary of the object's value.

PhyloDiversity objects can be plotted by `plot.PhyloValue` because PhyloDiversity objects are also of class PhyloValue.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Chao, A., Chiu, C.-H. and Jost, L. (2010). Phylogenetic diversity measures based on Hill numbers. *Philosophical Transactions of the Royal Society B* 365(1558): 3599-609.

Marcon, E., Herault, B. (2014). Decomposing Phylodiversity. *HAL* hal-00946177(version 1).

Pavoine, S., Love, M. S. and Bonsall, M. B. (2009). Hierarchical partitioning of evolutionary and ecological patterns in the organization of phylogenetically-structured species assemblages: Application to rockfish (genus: *Sebastes*) in the Southern California Bight. *Ecology Letters* 12(9): 898-908.

**See Also**

[PhyloEntropy, Diversity](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest
# and their taxonomy)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Calculate the phylogenetic Shannon diversity of the plot
```

```
summary(PhyloDiversity(Ps, 1, Paracou618.Taxonomy) -> d)
plot(d)

# Ns is the vector of abundances of the metacommunity
Ns <- Paracou618.MC$Ns
# Calculate the phylogenetic Shannon diversity of the plot
summary(bcPhyloDiversity(Ns, 1, Paracou618.Taxonomy, Correction = "Best") -> d)
plot(d)
```

---

PhyloEntropy

*Phylogenetic Entropy of a community*


---

### Description

Calculates the phylogenetic entropy of order  $q$  of a probability vector.

### Usage

```
PhyloEntropy(Ps, q = 1, Tree, Normalize = TRUE, CheckArguments = TRUE)
bcPhyloEntropy(Ns, q = 1, Tree, Normalize = TRUE, Correction = "Best",
  CheckArguments = TRUE)
is.PhyloEntropy(x)
## S3 method for class 'PhyloEntropy'
summary(object, ...)
```

### Arguments

Ps	A probability vector, summing to 1.
Ns	A numeric vector containing species abundances.
q	A number: the order of entropy. Default is 1.
Tree	An object of class <a href="#">hclust</a> or <a href="#">phylog</a> . The tree must be ultrametric.
Normalize	If TRUE (default), the Total entropy returned by the function is normalized by the height of the tree (it is the weighted average value of the entropy in each slice). If FALSE, it is the unnormalized weighted sum of the results.
Correction	A string containing one of the possible corrections: "None" (no correction), "ChaoShen", "Grassberger", "Holste", "Bonachela" or "Best", the default value.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.
x	An object to be tested or plotted
object	A PhyloDiversity object to be summarized.
...	Additional arguments to be passed to the generic methods.

**Details**

The phylogenetic entropy is its generalization of HCDT entropy to unequal species distances (Pavoine et al., 2009).

Calculation relies on [Tsallis](#) and [PhyloApply](#).

Intervals separate two cuts in a tree: no node is found at heights contained in an interval.

Bias correction requires the number of individuals to estimate sample [Coverage](#). Use `bcPhyloEntropy` and choose the `Correction`.

**Value**

An object of class `PhyloEntropy` is a list:

Distribution	The distribution used to calculate entropy
Function	The function used to calculate entropy
Tree	The functional or phylogenetic tree used to calculate entropy
Normalized	Logical. Indicates whether phyloentropy is normalized or proportional to the height of the tree.
Type	The type of entropy ("alpha", "beta" or "gamma").
Order	The order of entropy $q$ .
Cuts	A named vector containing values of neutral entropy along the tree. Names are cut ends, <i>i.e.</i> the ends of intervals (the first interval starts at 0 for leaves, the max value is the height of the tree).
Total	A value equal the total entropy multiplied by the tree height if <code>Normalize</code> is <code>FALSE</code> .

`is.PhyloEntropy` returns TRUE if the object is of class `PhyloEntropy`.

`summary.PhyloEntropy` returns a summary of the object's value.

`PhyloEntropy` objects can be plotted by `plot.PhyloValue` because `PhyloEntropy` objects are also of class `PhyloValue`.

**Author(s)**

Eric Marcon <[Eric.Marcon@ecofog.gf](mailto:Eric.Marcon@ecofog.gf)>

**References**

Marcon, E., Hérault, B. (2014). Decomposing PhyloDiversity. *HAL* hal-00946177(version 1).

Pavoine, S., Love, M. S. and Bonsall, M. B. (2009). Hierarchical partitioning of evolutionary and ecological patterns in the organization of phylogenetically-structured species assemblages: Application to rockfish (genus: *Sebastes*) in the Southern California Bight. *Ecology Letters* 12(9): 898-908.

**See Also**

[Tsallis](#), [PhyloDiversity](#)

**Examples**

```

# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest
# and their taxonomy)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Calculate the phylogenetic Shannon entropy of the plot
summary(PhyloEntropy(Ps, 1, Paracou618.Taxonomy) -> e)
plot(e)

# Ns is the vector of abundances of the metacommunity
Ns <- Paracou618.MC$Ns
# Calculate the phylogenetic Shannon entropy of the plot
summary(bcPhyloEntropy(Ns, 1, Paracou618.Taxonomy, Correction = "Best") -> e)
plot(e)

```

---

plot.PhyloValue	<i>Plot phylogenetic entropy of diversity.</i>
-----------------	------------------------------------------------

---

**Description**

Plots entropy or diversity against the height of the phylogenetic or functional tree.

**Usage**

```

## S3 method for class 'PhyloValue'
plot(x, xlab = expression(italic("T")), ylab = NULL, main = NULL, ...)

```

**Arguments**

x	An object of class PhyloValue, including <a href="#">PhyloDiversity</a> and <a href="#">PhyloEntropy</a> objects.
xlab	The X axis label, "T" by default for Time.
ylab	The Y axis label. if NULL (by default), "Entropy" or "Diversity" or nothing is chosen according to the object class.
main	The main title of the plot. if NULL (by default), a default value is used.
...	Additional arguments to be passed to <a href="#">plot</a> .

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>



---

PPtree                      *Preprocessed Trees.*

---

## Description

Methods for objects of type "PPtree".

## Usage

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

## Arguments

x                      An object to be tested or plotted  
...                    Additional arguments to be passed to the generic methods.

## Value

An object of class **PPtree** is a list:

phyTree	A <a href="#">phylog</a> tree
hTree	A <a href="#">hclust</a> tree
Height	The height of the tree, that is to say the distance between root and leaves
Cuts	A vector. Cut times of the tree (the distance from nodes to leaves)
Intervals	A vector. The lengths of intervals between cuts

is.PPtree returns TRUE if the object is of class PPtree.

plot.PPtree plots it.

## Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf>

---

Rao

*Rao Quadratic Entropy of a Community*

---

### Description

Calculates Rao's quadratic entropy of a community described by a probability vector and a phylogenetic / functional tree.

### Usage

```
Rao(Ps, Tree, CheckArguments = TRUE)
```

### Arguments

**Ps** A probability vector, summing to 1.

**Tree** An object of class `hclust` or `phylog`. The tree must be ultrametric.

**CheckArguments** Logical; 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 is not possible when diversity is calculated from probabilities.

Use `bcPhyloDiversity(Ps, 2, Tree)` to do it.

### Value

A number equal to the calculated diversity.

### Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf>

### References

Rao, C. R. (1982). Diversity and dissimilarity coefficients: a unified approach. *Theoretical Population Biology* 21(24-43).

### See Also

[bcPhyloDiversity](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest
# and their taxonomy)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Calculate the phylogenetic Shannon diversity of the plot
Rao(Ps, Paracou618.Taxonomy)
```

---

Richness

*Simpson entropy of a community*

---

**Description**

Calculates the richness of a community (described by a vector of abundances or probabilities).

**Usage**

```
Richness(Ns, CheckArguments = TRUE)
```

**Arguments**

**Ns** A numeric vector. Contains either abundances or probabilities.

**CheckArguments** Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

Richness simply returns the number of strictly positive elements of the vector.

**Value**

A number equal to the calculated entropy.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**See Also**

[Tsallis](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Richness of the first community of Paracou618.MC
Richness(Paracou618.MC$Psi[, 1])
```

---

Shannon *Shannon entropy of a community*

---

### Description

Calculates the Shannon entropy of a probability vector.

### Usage

```
Shannon(Ps, CheckArguments = TRUE)
bcShannon(Ns, Correction = "Best", CheckArguments = TRUE)
```

### Arguments

Ps	A probability vector, summing to 1.
Ns	A numeric vector containing species abundances.
Correction	A string containing one of the possible corrections: "None" (no correction), "ChaoShen", "Grassberger", "Grassberger2003", "Schurmann", "Holste", "Bonachela", "ZhangHz", "ChaoWangJost" or "Best", the default value. Currently, "Best" is "ChaoWangJost".
CheckArguments	Logical; 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 to estimate sample [Coverage](#). Use `bcShannon` and choose the `Correction`.

Correction techniques are from Chao and Shen (2003), Grassberger (1988), Grassberger (2003), Schurmann (2003), Holste *et al.* (1998), Bonachela *et al.* (2008), Zhang (2012), Chao, Wang and Jost (2013).

Using [MetaCommunity](#) mutual information, Chao, Wang and Jost (2013) calculate unbiased Shannon beta entropy (see the last example below) with better results than the Chao and Shen estimator, but community weights cannot be arbitrary: they must be proportional to the number of individuals.

### Value

A number equal to the calculated entropy.

### Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf>

## References

- Bonachela, J. A., Hinrichsen, H. and Munoz, M. A. (2008). Entropy estimates of small data sets. *Journal of Physics A: Mathematical and Theoretical* 41(202001): 1-9.
- Chao, A. and Shen, T. J. (2003). Nonparametric estimation of Shannon's index of diversity when there are unseen species in sample. *Environmental and Ecological Statistics* 10(4): 429-443.
- Chao, A., Wang, Y. T. and 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.
- Grassberger, P. (1988). Finite sample corrections to entropy and dimension estimates. *Physics Letters A* 128(6-7): 369-373.
- Grassberger, P. (2003). Entropy Estimates from Insufficient Samplings. *ArXiv Physics e-prints* 0307138.
- Holste, D., Grosse, I. and Herzel, H. (1998). Bayes' estimators of generalized entropies. *Journal of Physics A: Mathematical and General* 31(11): 2551-2566.
- Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal* 27: 379-423, 623-656.
- Schurmann, T. (2004). Bias analysis in entropy estimation. *Journal of Physics A: Mathematical and Theoretical* 37(27): L295-L301.
- Tsallis, C. (1988). Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics* 52(1): 479-487.
- Zhang, Z. (2012). Entropy Estimation in Turing's Perspective. *Neural Computation* 24(5): 1368-1389.

## See Also

[bcShannon](#), [Tsallis](#)

## Examples

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Calculate Shannon entropy
Shannon(Ps)

# Ns is the vector of abundances of the metacommunity
Ns <- Paracou618.MC$Ns
# Calculate an unbiased estimator of Shannon entropy
bcShannon(Ns)

# Use metacommunity data to calculate unbiased Shannon beta as mutual information
(bcShannon(Paracou618.MC$Ns) + bcShannon(colSums(Paracou618.MC$Nsi))
 - bcShannon(Paracou618.MC$Nsi))
# Compare with Chao and Shen correction (works for Tsallis entropy in general)
summary(BetaEntropy(Paracou618.MC, 1))
```

ShannonBeta

*Shannon beta entropy of a community***Description**

Calculates the Shannon beta entropy of a community belonging to a metacommunity.

**Usage**

```
ShannonBeta(Ps, Pexp, CheckArguments = TRUE)
bcShannonBeta(Ns, Nexp, Correction = "Best", CheckArguments = TRUE)
```

**Arguments**

Ps	The probability vector of species of the community.
Pexp	The probability vector of species of the metacommunity.
Ns	A numeric vector containing species abundances of the community.
Nexp	A numeric vector containing species abundances of the metacommunity.
Correction	A string containing one of the possible corrections: currently, only "ChaoShen", identical to "Best".
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

The derivation of Shannon beta entropy can be found in Marcon et al. (2012).

Bias correction requires the number of individuals to estimate sample [Coverage](#). Use bcShannonBeta and choose the Correction.

**Value**

A number equal to the calculated entropy.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Marcon, E., Herault, B., Baraloto, C. and Lang, G. (2012). The Decomposition of Shannon's Entropy and a Confidence Interval for Beta Diversity. *Oikos* 121(4): 516-522.

**See Also**

[bcShannonBeta](#)

**Examples**

```

# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Probability distribution of the first plot
Ps1 <- Paracou618.MC$Psi[, 1]
# Shannon beta entropy of the plot
ShannonBeta(Ps1, Ps)

# Ns is the vector of abundances of the metacommunity
Ns <- Paracou618.MC$Ns
# Abundances in the first plot
Ns1 <- Paracou618.MC$Nsi[, 1]
# Unbiased Shannon beta entropy of the plot
bcShannonBeta(Ns1, Ns)

```

---

Simpson	<i>Simpson entropy of a community</i>
---------	---------------------------------------

---

**Description**

Calculates the Simpson entropy of a probability vector.

**Usage**

```

Simpson(Ps, CheckArguments = TRUE)
bcSimpson(Ns, Correction = "Lande", CheckArguments = TRUE)

```

**Arguments**

Ps	A probability vector, summing to 1.
Ns	A numeric vector containing species abundances.
Correction	A string containing one of the possible corrections accepted by <a href="#">bcTsallis</a> or "Lande", the default value.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

**Details**

Lande's correction has been derived (Lande, 1996; Good, 1953) especially for Simpson entropy, while other corrections are for generalized Tsallis entropy. It is identical to the unbiased estimator proposed by Simpson, although arguments were different. Bias correction requires the number of individuals to estimate sample [Coverage](#). Use `bcSimpson` and choose the `Correction`.

**Value**

A number equal to the calculated entropy.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**References**

Good, I. J. (1953). On the Population Frequency of Species and the Estimation of Population Parameters. *Biometrika* 40(3/4): 237-264.

Lande, R. (1996). Statistics and partitioning of species diversity, and similarity among multiple communities. *Oikos* 76: 5-13.

Simpson, E. H. (1949). Measurement of diversity. *Nature* 163(4148): 688.

**See Also**

[Tsallis](#), [bcSimpson](#)

**Examples**

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Calculate Simpson entropy
Simpson(Ps)

# Ns is the vector of abundances of the metacommunity
Ns <- Paracou618.MC$Ns
# Calculate an unbiased estimator of Simpson's index of diversity
bcSimpson(Ns)
```

---

SimpsonBeta

*Simpson beta entropy of a community*

---

**Description**

Calculates the Simpson beta entropy of a community belonging to a metacommunity.

**Usage**

```
SimpsonBeta(Ps, Pexp, CheckArguments = TRUE)
bcSimpsonBeta(Ns, Nex, Correction = "Best", CheckArguments = TRUE)
```

**Arguments**

Ps	The probability vector of species of the community.
Pexp	The probability vector of species of the metacommunity.
Ns	A numeric vector containing species abundances of the community.
Nexp	A numeric vector containing species abundances of the metacommunity.



- Correction A string containing one of the possible corrections: currently, only "ChaoShen", identical to "Best".
- CheckArguments Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

### Details

The derivation of Tsallis beta entropy (Simpson is Tsallis of order 2) can be found in Marcon et al. (2014).

Bias correction requires the number of individuals to estimate sample [Coverage](#). Use `bcSimpsonBeta` and choose the `Correction`.

Note that Simpson beta entropy value is related to Simpson alpha entropy value and cannot be compared across communities (Jost, 2007). Beta entropy of a community is not meaningful in general, do rather calculate the [BetaDiversity](#) of order 2 of the metacommunity.

### Value

A number equal to the calculated entropy.

### Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf>

### References

- Jost (2007), Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.
- Marcon, E., Scotti, I., Hérault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.

### See Also

[Simpson](#), [bcSimpsonBeta](#), [BetaDiversity](#)

### Examples

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Probability distribution of the first plot
Ps1 <- Paracou618.MC$Psi[, 1]
# Shannon beta entropy of the plot
SimpsonBeta(Ps1, Ps)
# Transform into diversity
expq(SimpsonBeta(Ps1, Ps)/(1-Simpson(Ps1)), 2)

# Ns is the vector of abundances of the metacommunity
Ns <- Paracou618.MC$Ns
# Abundances in the first plot
```

```

Ns1 <- Paracou618.MC$Nsi[, 1]
# Unbiased Shannon beta entropy of the plot
bcSimpsonBeta(Ns1, Ns)

```

---

SimTest

*SimTest class*


---

### Description

Methods for objects of type "SimTest", used to test a value against its distribution under a simulated null hypothesis.

### Usage

```

as.SimTest(RealValue, SimulatedValues)
is.SimTest(x)
## S3 method for class 'SimTest'
plot(x, Quantiles = c(0.025, 0.975), ...,
     colValue = "red", lwdValue = 2, ltyValue = 2,
     colQuantiles = "black", lwdQuantiles = 1, ltyQuantiles = 2)
## S3 method for class 'SimTest'
summary(object, Quantiles = c(0.025, 0.975), ...)

```

### Arguments

RealValue	A numeric Value (the actual one).
SimulatedValues	A numeric vector containing the simulated values.
x	An object to be tested or plotted.
object	A SimTest object to be summarized.
Quantiles	A vector containing the quantiles of interest.
colValue	The color of the line representing the real value on the plot.
lwdValue	The width of the line representing the real value on the plot.
ltyValue	The line type of the line representing the real value on the plot.
colQuantiles	The color of the lines representing the quantiles on the plot.
lwdQuantiles	The width of the lines representing the quantiles on the plot.
ltyQuantiles	The line type of the lines representing the quantiles on the plot.
...	Additional arguments to be passed to the generic methods.

### Details

Simulated values should be obtained by simulation. The actual value is compared to simulated quantiles. SimTest objects can be plotted and summarized.

**Value**

SimTest objects are lists containing:

RealValue        The value to test.

SimulatedValues

A vector of simulated values, whose quantiles will be used for the test.

is.SimTest returns TRUE if the object is of class SimTest.

summary.SimTest returns a summary of the object, including the empirical quantile of the real value in the simulated distributon.

**Author(s)**

Eric Marcon <Eric.Marcon@ecofog.gf>

**Examples**

```
# Set the value to test
Real <- 0.8
# Is it a realization of a Gaussian distribution?
Sims <- rnorm(1000)
# Make a Simtest object
st <- as.SimTest(Real, Sims)
plot(st)
summary(st)
```

---

Tsallis

*Tsallis Entropy of a community*

---

**Description**

Calculates the HCDT, also known as Tsallis entropy of order  $q$  of a probability vector.

**Usage**

```
Tsallis(Ps, q = 1, CheckArguments = TRUE)
bcTsallis(Ns, q = 1, Correction = "Best", CheckArguments = TRUE)
```

**Arguments**

Ps	A probability vector, summing to 1.
Ns	A numeric vector containing species abundances.
q	A number: the order of entropy. Some corrections allow only a positive number. Default is 1 for Shannon entropy.
Correction	A string containing one of the possible corrections: "None" (no correction), "ChaoShen", "Grassberger", "Holste", "Bonachela", "ZhangGrabchak" or "Best", the default value.
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

### Details

Tsallis (Havrda and Charvat, 1967; Daroczy, 1970; Tsallis, 1988) generalized entropy is a generalized measure of diversity (Jost, 2006).

Bias correction requires the number of individuals to estimate sample Coverage. Use bcTsallis and choose the Correction. Correction techniques are from Chao and Shen (2003), Grassberger (1988), Holste *et al.* (1998), Bonachela *et al.* (2008) and Zhang and Grabchak (2014).

Currently, the "Best" correction is the max value of "ChaoShen" and "Grassberger" (Marcon *et al.*, 2014).

### Value

A number equal to the calculated entropy.

### Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf>

### References

Havrda, J. and Charvat, F. (1967). Quantification method of classification processes. Concept of structural a-entropy. *Kybernetika* 3(1): 30-35.

Daroczy, Z. (1970). Generalized information functions. *Information and Control* 16(1): 36-51.

Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.

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

Jost, L. (2006). Entropy and diversity. *Oikos* 113(2): 363-375.

Zhang, Z. and Grabchak M. (2014). Entropic Representation and Estimation of Diversity Indices. *arXiv* 1403.3031(v. 1): 1-12.

### Examples

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Calculate Tsallis entropy of order 2, i.e. Simpson's index of diversity
Tsallis(Ps, 2)

# Ns is the vector of abundances of the metacommunity
Ns <- Paracou618.MC$Ns
# Calculate an unbiased estimator of Tsallis entropy of order 2, i.e. Simpson's index
bcTsallis(Ns, 2)
```

---

TsallisBeta

*Tsallis beta entropy of a community*


---

### Description

Calculates the Tsallis beta entropy of order  $q$  of a community belonging to a metacommunity.

### Usage

```
TsallisBeta(Ps, Pexp = NULL, q, CheckArguments = TRUE)
bcTsallisBeta(Ns, Nexp = NULL, q, Correction = "Best", CheckArguments = TRUE)
```

### Arguments

Ps	The probability vector of species of the community.
Pexp	The probability vector of species of the metacommunity.
Ns	A numeric vector containing species abundances of the community.
Nexp	A numeric vector containing species abundances of the metacommunity.
q	A number: the order of entropy. Default is 1 for Shannon entropy.
Correction	A string containing one of the possible corrections: currently, only "ChaoShen" or "None". "Best" is the default value, it is equivalent to "ChaoShen".
CheckArguments	Logical; if TRUE, the function arguments are verified. Should be set to FALSE to save time when the arguments have been checked elsewhere.

### Details

The derivation of Tsallis beta entropy can be found in Marcon et al. (2014).

Bias correction requires the number of individuals to estimate sample [Coverage](#). Use `bcTsallisBeta` and choose the `Correction`.

Note that beta entropy value is related to alpha entropy (if  $q$  is not 1) and cannot be compared across communities (Jost, 2007). Beta entropy of a community is not meaningful in general, do rather calculate the [BetaDiversity](#) of the metacommunity.

### Value

A number equal to the calculated entropy.

### Author(s)

Eric Marcon <Eric.Marcon@ecofog.gf>

## References

Jost (2007), Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.

Marcon, E., Scotti, I., Herault, B., Rossi, V. and Lang, G. (2014). Generalization of the partitioning of Shannon diversity. *PLOS One* 9(3): e90289.

## Examples

```
# Load Paracou data (number of trees per species in two 1-ha plot of a tropical forest)
data(Paracou618)
# Ps is the vector of probabilities
Ps <- Paracou618.MC$Ps
# Probability distribution of the first plot
Ps1 <- Paracou618.MC$Psi[, 1]
# Divergence of order 2 between plot 1 and the whole forest
TsallisBeta(Ps1, Ps, 2)

# Ns is the vector of abundances of the metacommunity
Ns <- Paracou618.MC$Ns
# Abundances in the first plot
Ns1 <- Paracou618.MC$Nsi[, 1]
# Divergence of order 2 between plot 1 and the whole forest, with bias correction
bcTsallisBeta(Ns1, Ns, 2)
```

# Index

## \*Topic **datasets**

EightSpAbundance, 25  
EightSpTree, 25  
Paracou618.dist, 44  
Paracou618.Functional, 45  
Paracou618.MC, 46  
Paracou618.Taxonomy, 46

## \*Topic **package**

entropart-package, 3

AllenH, 4, 12

AlphaDiversity, 6

AlphaEntropy, 6, 7, 7, 19, 21

as.SimTest (SimTest), 66

bcDiversity (Diversity), 15

bcDqz, 4

bcDqz (Dqz), 23

bcHqz, 8, 31

bcHqz (Hqz), 32

bcHqzBeta, 10

bcHqzBeta (HqzBeta), 33

bcPhyloBetaEntropy, 10, 51

bcPhyloBetaEntropy (PhyloBetaEntropy),  
50

bcPhyloDiversity, 4, 47, 48, 58

bcPhyloDiversity (PhyloDiversity), 52

bcPhyloEntropy, 8, 27, 31

bcPhyloEntropy (PhyloEntropy), 54

bcShannon, 4, 27, 61

bcShannon (Shannon), 60

bcShannonBeta, 62

bcShannonBeta (ShannonBeta), 62

bcSimpson, 27, 64

bcSimpson (Simpson), 63

bcSimpsonBeta, 65

bcSimpsonBeta (SimpsonBeta), 64

bcTsallis, 8, 27, 31, 63

bcTsallis (Tsallis), 67

bcTsallisBeta, 9–11

bcTsallisBeta (TsallisBeta), 69

beta, 26

BetaDiversity, 8, 10, 11, 34, 65, 69

BetaEntropy, 9, 10, 19, 21

ChaoPD, 5, 11, 13

CommunityProfile, 13

Coverage, 14, 16, 47, 50, 52, 55, 60, 62, 63,  
65, 68, 69

dist, 44

Diversity, 4, 15, 53

DivEst, 4, 16

DivPart, 4, 17, 18, 18, 23

DivProfile, 4, 20, 21

Dqz, 4, 23, 24, 32, 33

EightSpAbundance, 25

EightSpTree, 25

Enq, 26

entropart (entropart-package), 3

entropart-package, 3

EntropyCI, 27

expq, 16, 28, 29, 37

GammaDiversity, 29

GammaEntropy, 19, 21, 29, 30, 30

hclust, 4–7, 9, 10, 12, 17, 19, 21, 29, 31, 43,  
45, 47, 48, 50, 52, 54, 57, 58

Hqz, 24, 32

HqzBeta, 33

is.MCdiversity (MCdiversity), 38

is.MCentropy (MCentropy), 40

is.MetaCommunity (MetaCommunity), 41

is.PhyloDiversity (PhyloDiversity), 52

is.PhyloEntropy (PhyloEntropy), 54

is.PPtree (PPtree), 57

is.SimTest (SimTest), 66

- KLq, 35
- list, 38
- lnq, 36
- MC Utilities, 37
- MCdiversity, 6, 9, 38
- MCEntropy, 8, 11, 40
- MergeC (MC Utilities), 37
- MergeMC (MC Utilities), 37
- MetaCommunity, 4, 6, 7, 9, 10, 17, 19, 21, 22, 29, 31, 38–40, 41, 46, 60
- originality, 43
- Originality.Species, 43
- Paracou618.dist, 44, 45
- Paracou618.Functional, 45
- Paracou618.MC, 46
- Paracou618.Taxonomy, 46
- PDFD, 47
- PhyloApply, 48, 50, 55
- PhyloBetaEntropy, 50
- PhyloDiversity, 12, 24, 49–51, 52, 55, 56
- PhyloEntropy, 4, 5, 33, 49, 51, 53, 54, 56
- phylog, 4–7, 9, 10, 12, 17, 19, 21, 25, 29, 31, 43, 47, 48, 50, 52, 54, 57, 58
- plot, 56
- plot.DivEst (DivEst), 16
- plot.DivPart (DivPart), 18
- plot.DivProfile (DivProfile), 21
- plot.MCdiversity (MCdiversity), 38
- plot.MCEntropy (MCEntropy), 40
- plot.MetaCommunity (MetaCommunity), 41
- plot.PhyloValue, 53, 55, 56
- plot.PPtree (PPtree), 57
- plot.SimTest (SimTest), 66
- PPtree, 4, 5, 12, 57
- Preprocess.Tree, 49
- Rao, 4, 58
- Richness, 59
- Shannon, 60
- ShannonBeta, 62
- ShuffleMC (MC Utilities), 37
- Simpson, 63, 65
- SimpsonBeta, 64
- SimTest, 66
- summary.DivEst (DivEst), 16
- summary.DivPart (DivPart), 18
- summary.DivProfile (DivProfile), 21
- summary.MCdiversity (MCdiversity), 38
- summary.MCEntropy (MCEntropy), 40
- summary.MetaCommunity (MetaCommunity), 41
- summary.PhyloDiversity (PhyloDiversity), 52
- summary.PhyloEntropy (PhyloEntropy), 54
- summary.SimTest (SimTest), 66
- Tsallis, 13, 16, 55, 59, 61, 64, 67
- TsallisBeta, 36, 50, 51, 69