entropart: An R Package to Measure and Partition Diversity

Eric Marcon
AgroParisTech
UMR EcoFoG

Bruno Hérault
Cirad
UMR EcoFoG

Abstract

entropart is a package for R designed to estimate diversity based on HCDT entropy or similarity-based entropy.

It allows calculating species-neutral, phylogenetic and functional entropy and diversity, partitioning them and correcting them for estimation bias.

Keywords: biodiversity, entropy, partitioning.

1. Introduction

Diversity measurement can be done through a quite rigorous framework based on entropy, *i.e.* the amount of uncertainty calculated from the frequency distribution of a community (???). Tsallis entropy, also known as HCDT entropy (???), is of particular interest (??) namely because it gathers the number of species, Shannon (?) and Simpson (?) indices of diversity into a single framework. Interpretation of entropy is not straightforward but one can easily transform it into Hill numbers (?) which have many desirable properties (?): mainly, they are the number of equally-frequent species that would give the same level of diversity as the data.

? generalized the duality of entropy and diversity, deriving the relation between phylogenetic or functional diversity (?) and phylogenetic or functional entropy (we will write *phylodiversity* and *phyloentropy* for short), as introduced by ?. Special cases are the well-known PD (?) and FD (?) indices and Rao's (?) quadratic entropy. The same relation holds between Ricotta and Szeidl entropy of a community (?) and similarity-based diversity (?).

The **entropart** package for R (?) enables calculation of all these measures of diversity and entropy and their partitioning.

Diversity partitioning means that, in a given area, the γ diversity D_γ of all individuals found may be split into within (α diversity, D_α) and between (β diversity, D_β) local assemblages. α diversity reflects the diversity of individuals *in* local assemblages whereas β diversity reflects the diversity *of* the local assemblages. ? derived the decomposition of Tsallis γ entropy into its α and β components, generalized to phylodiversity (?) and similarity-based diversity (?).

Estimators of diversity are biased because of unseen species and also because they are not linear functions of probabilities (?). α and γ diversities are underestimated by naive estimators (??). β diversity is severely biased too when sampling is not sufficient (?). Bias-corrected estimators of phylodiversity have been developed by **?**. Estimators of similarity-based diversity were derived by **?**. The package includes them all.

In summary, the framework supported by the package is as follows. First, an information function is chosen to describe the amount of surprise brought by the observation of each individual. In the simplest case of species-neutral diversity, it is just a decreasing function of probability: observing an individual of a rarer species brings more surprise. Various information functions allow evaluating species-neutral, phylogenetic or functional entropy. Surprise is averaged among all individuals of a community to obtain its entropy. Entropy is systematically transformed into diversity for interpretation. Diversity is an effective number of species, *i.e.* the number of equally-different and equally-frequent species that would give the same entropy as the data. The average entropy of communities of an assemblage is α entropy, while the entropy of the assemblage is γ entropy. Their difference is β entropy. After transformation, β diversity is the ratio of γ to α diversity. It is an effective number of communities, *i.e.* the number of equally-weighted communities with no species in common necessary to obtain the same diversity as the data. Estimation-bias correction is more easily applied to entropy before transforming it into diversity.

This framework is somehow different from that of **?** who define α diversity in another way (see **?**, for a detailed comparison), such that α entropy is not the average surprise of an assemblage. They also propose a definition of functional diversity (**?**) based in the information brought by pairs of individuals that is not supported in the package.

The successive sections of this paper presents the package features, illustrated by worked examples based on the data included in the package.

2. Package organization

2.1. Data

Most functions of the package calculate entropy or diversity of a community or of an assemblage of communities called a "meta-community". Community functions accept a vector of probabilities or of abundances for species data. Each element of the vector contains the probability or the number of occurrences of a species in a given community. Meta-community functions require a particular data organization in a `MetaCommunity` object described here.

A `MetaCommunity` is basically a list. Its main components are `$Nsi`, a matrix containing the species abundances whose lines are species, columns are communities and `$Wi`, a vector containing community weights. Creating a `MetaCommunity` object is the purpose of the `MetaCommunity` function. Arguments are a dataframe containing the number of individuals per species (lines) in each community (columns), and a vector containing the community weights. The following example creates a `MetaCommunity` made of three communities of unequal weights with 4 species. The weighted average probabilities of occurrence of species and the total number of individuals define the meta-community as the assemblage of communities.

```
R> library("entropart")
R> (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")))
```

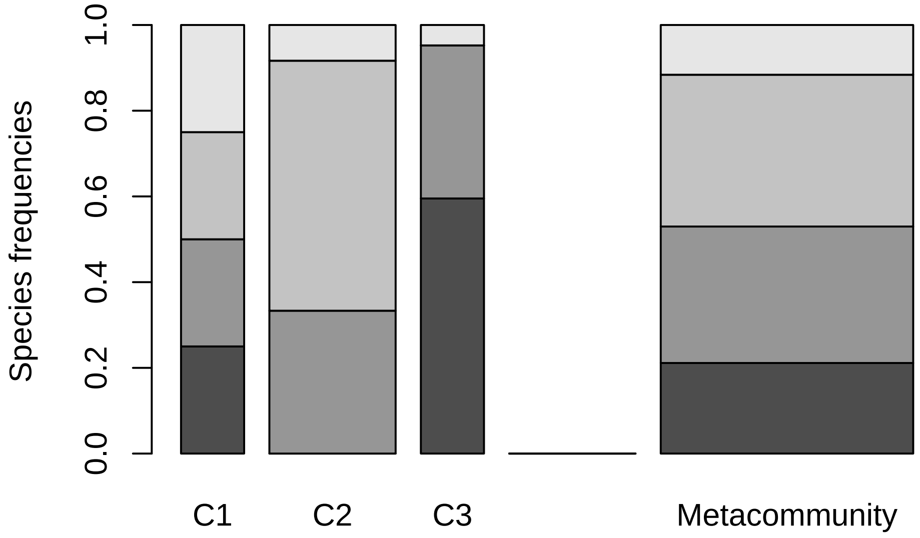

Figure 1. Plot of a `MetaCommunity`. Communities (named C1, C2 ad C3) are represented in the left part of the figure, the metacommunity to the right. Bar widths are proportional to community weights. Species abundances are represented vertically (4 species are present in the meta-community, only 3 of them in communities C2 and C3.

```
    C1 C2 C3
sp1 10  0 25
sp2 10 20 15
sp3 10 35  0
sp4 10  5  2
```

```
R> w <- c(1, 2, 1)
R> MC <- MetaCommunity(Abundances = df, Weights = w)
```

A meta-community is partitioned into several local communities (indexed by $i = 1,\ 2, \dots, I$). n_i individuals are sampled in community i. Let $s = 1,\ 2, \dots, S$ denote the species that compose the meta-community, $n_{s,i}$ the number of individuals of species s sampled in the local community i, $n_s = \sum_i n_{s,i}$ the total number of individuals of species s, $n = \sum_s \sum_i n_{s,i}$ the total number of sampled individuals. Within each community i, the probability $p_{s,i}$ for an individual to belong to species s is estimated by $\hat{p}_{s,i} = n_{s,i}/n_i$. The same probability for the meta-community is p_s. Communities have a weight w_i, satisfying $p_s = \sum_i w_i p_{s,i}$. The commonly-used $w_i =\ n_i/n$ is a possible weight, but the weighting may be arbitrary (*e.g.* the sampled areas). The component `$Ps` of a `MetaCommunity` object contains the probability of occurrence of each species in the meta-community, calculated this way:

```
R> MC$Ps
```

```
      sp1       sp2       sp3       sp4 
0.2113095 0.3184524 0.3541667 0.1160714 
```

A `MetaCommunity` can be summarized and plotted (Figure 1).

The package contains an example dataset containing the inventory of two 1-ha tropical forest plots in Paracou, French Guiana (?):

```
R> data("Paracou618")
R> summary(Paracou618.MC)
```

```
Meta-community (class 'MetaCommunity') made of  1124 individuals in 2
communities and 425 species.

Its sample coverage is  0.92266748426447

Community weights are:
[1] 0.5720641 0.4279359
Community sample numbers of individuals are:
P006 P018
 643  481
Community sample coverages are:
     P006      P018
0.8943859 0.8463782
```

`Paracou618.MC` is a meta-community made of two communities named "P006" and "P018", containing 425 species (their name is *Family_Genus_Species*, abbreviated to 4 characters). The values of the abundance matrix are the number of individuals of each species in each community. Sample coverage will be explained later.

The dataset also contains a taxonomy and a functional tree. `Paracou618.Taxonomy` is an object of class `phylog`, defined in **ade4** (?), namely a phylogenetic tree. This example data is only a taxonomy, containing family, genus and species levels for the sake of simplicity. `Paracou618.Functional` is an object of class `hclust` containing a functional tree based on leaf, height, stem and seed functional traits (??). The package accepts any ultrametric tree of class `phylog` or `hclust`. `Paracou618.dist` is the distance matrix (actually a `dist` object) used to build the functional tree.

Numeric vectors containing species abundances (such as the `$Ns` component of `MetaCommunity`) or probabilities (such as `$Ps`) may be converted to abundance vectors (`AbdVector`) or probability vectors (`ProbaVector`) to clarify their content. By default, the `as.AbdVector` function transforms abundance values into integer if they are not (the `$Ns` components of a `MetaCommunity` is typically not an integer vector if community weights are not proportional to their numbers of individuals):

```
R> data("Paracou618")
R> PAbd <- as.AbdVector(Paracou618.MC$Ns)
```

The `as.ProbaVector` function transforms abundances to probabilities if necessary:

```
R> PProba <- as.ProbaVector(Paracou618.MC$Ns)
```

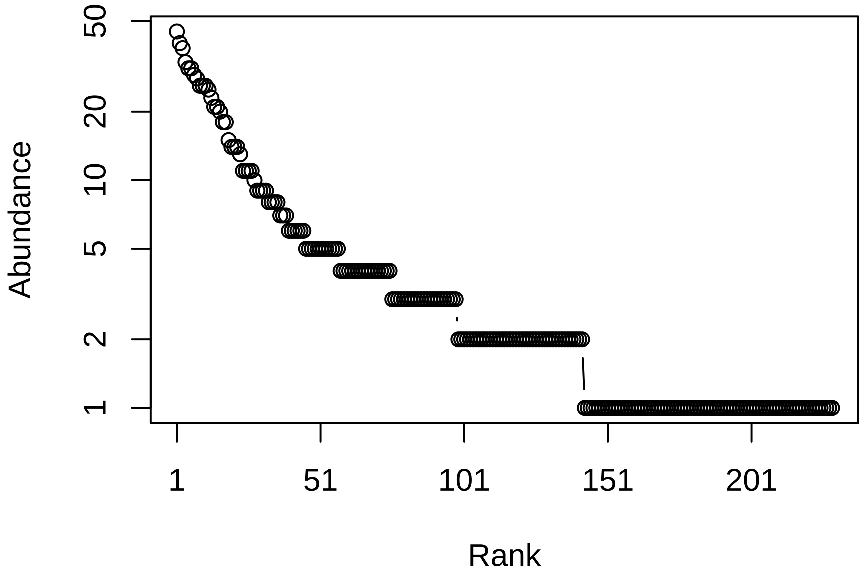

Figure 2. Plot of a `ProbaVector`. Rank-Abundance curve (Whittaker plot) of the Paracou species distribution.

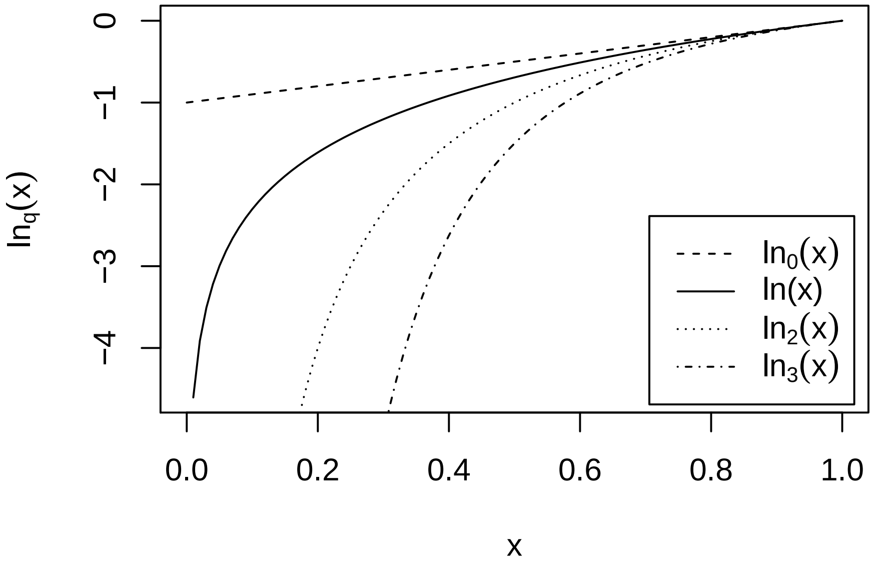

Figure 3. Curves of $\ln_q x$ for different values of q between 0 and 4 ($\ln_1 x = \ln x$).

`AbdVector` and `ProbaVector` objects both are `SpeciesDistribution` objects which can be plotted (Figure 2).

2.2. Utilities

The deformed logarithm formalism (**?**) is very convenient to manipulate entropies. The deformed logarithm of order q is defined as:

$$\ln_q x = \frac{x^{1-q} - 1}{1 - q} \tag{1}$$

It converges to ln when $q \to 1$, see figure 3.

The inverse function of $\ln_q x$ is the deformed exponential:

$$e_q^x = [1 + (1 - q)x]^{\frac{1}{1-q}} \tag{2}$$

Functions of the packages are `lnq(x, q)` and `expq(x, q)`.

3. Species-neutral diversity

3.1. Community functions

HCDT entropy

Species-neutral HCDT entropy of order q of a community is defined as:

$$^{q}H = \frac{1 - \sum_{s} p_s^q}{q-1} = -\sum_{s} p_s^q \ln_q p_s \tag{3}$$

`q` is the order of diversity (*e.g.*: 1 for Shannon). Entropy can be calculated by the **Tsallis** function. Paracou meta-community entropy of order 1 is:

```
R> Tsallis(Ps = Paracou618.MC$Ps, q = 1)
```

```
[1] 4.736023
```

For convenience, special cases of entropy of order q have a clear-name function: `Richness` for $q = 0$, `Shannon` for $q = 1$, `Simpson` for $q = 2$.

```
R> Shannon(Ps = Paracou618.MC$Ps)
```

```
[1] 4.736023
```

Entropy values have no intuitive interpretation in general, except for the number of species ^{0}H and Simpson entropy ^{2}H which is the probability for two randomly chosen individuals to belong to different species.

Sample coverage

A useful indicator of sampling quality is the sample coverage (**???**), that is to say the probability for a species of the community to be observed in the actual sample. It equals the sum of the probability of occurrences of all observed species. Its historical estimator is (**?**):

$$\hat{C} = 1 - \frac{S^1}{n} \tag{4}$$

S^1 is the number of singletons (species observed once) of the sample, and n is its size. The estimator has been improved by taking into account the whole distribution of species (**?**). The `Coverage` function calculates it, allowing to choose the estimator (Zhang and Huang's by default):

```
R> Coverage(Ns = Paracou618.MC$Ns)
```

```
[1] 0.9220438
```

The sample coverage cannot be estimated from probability data: abundances are required.

Its interpretation is straightforward: some species have not been sampled. Their number is unknown but their total probability of occurence can be estimated accurately. Here, it is a bit less than 8%. From another point of view, the probability for an individual of the community to belong to a sampled species is C: 8% of them belong to missed species. The number of missed species may be estimated by `bcRichness` but this is not the point here. The sample coverage is the foundation of many estimators of entropy.

Estimation-bias corrected estimators

Estimation-bias correction is used to improve the estimation of entropy despite unobserved species and also mathematical issues (**?**). Bias-corrected estimators (often relying on sample coverage) are returned by functions whose names are prefixed by `bc`, such as `bcTsallis`. They are similar to the non-corrected ones but they use abundance data and propose several bias-correction techniques to select in the `Correction` argument. A "Best" correction is calculated by default, detailed in the help file of each function.

```
R> bcTsallis(Ns = Paracou618.MC$Ns, q = 1)
```

```
[1] 4.892159
```

The best correction for Tsallis entropy follows **?**. It combines an unbiased estimator previously derived by **?** and an estimate of the remaining bias.

All community functions such as `Tsallis` are actually generic methods that can handle several types of data the appropriate way: if the first argument of the function is a `ProbaVector` (or a numeric vector summing to 1), no bias correction is applied. If it is an `AbdVector` (or an integer vector), the bias-corrected estimator is used (*e.g.* `bcTsallis`).

The different ways to use the functions are a matter of personal preference. `bcTsallis` with a numeric vector:

```
R> bcTsallis(Paracou618.MC$Ns, q = 1)
```

```
[1] 4.892159
```

is equivalent to `Tsallis` with an abundance vector:

```
R> Tsallis(PAbd, q = 1)
```

```
[1] 4.892159
```

whilst `Tsallis` with a probability vector does not allow bias correction:

```
R> Tsallis(PProba, q = 1)
```

```
[1] 4.736023
```

Bias-corrected entropy is ready to be transformed into explicit diversity.

Effective numbers of species

Entropy should be converted into "true diversity" (?), *i.e.* effective number of species equal to ? numbers:

$$^{q}D = \left(\sum_{s} p_s^q \right)^{\frac{1}{1-q}} \tag{5}$$

This can be done by the deformed exponential function, or using directly the `Diversity` or `bcDiversity` functions (equal to the deformed exponential of order q of `Tsallis` or `bcTsallis`)

```
R> expq(Simpson(Ps = Paracou618.MC$Ps), q = 2)

[1] 68.7215

R> Diversity(Ps = Paracou618.MC$Ps, q = 2)

[1] 68.7215

R> expq(bcTsallis(Ns = Paracou618.MC$Ns, q = 2), q = 2)

[1] 73.13163

R> bcDiversity(Ns = Paracou618.MC$Ns, q = 2)

[1] 73.13163
```

The effective number of species of the Paracou dataset is estimated to be 73 after bias correction (rather than 69 without it). It means that a community made of 73 equally-frequent species has the same Simpson entropy as the actual one. This is much less than the actual 425 sampled species but Simpson's entropy focuses on dominant species.

Hurlbert's diversity

Hurlbert's index of diversity (?) of order k is the expected number of species observed in a sample of size k.

$$_{k}S = \sum_{s} \left[1 - (1 - p_s)^k \right] \tag{6}$$

Greater values of k give more importance to rare species.

An unibiased estimator of $_{k}S$ has been provided by Hurlbert, for values of k up to the sample size n:

$$_{k}\hat{S} = \sum_{s} \left[1 - \binom{n - n_s}{k} / \binom{n}{k} \right] \tag{7}$$

The effective number of species ${}_{k}D$ can be found by solving the following equation (**?**):

$$ {}_{k}S = {}_{k}D \left[1 - \left(1 - \frac{1}{{}_{k}D} \right)^{k} \right] \tag{8} $$

Hurlbert's index is calculated by the `Hurlbert` function, its unbiased estimator by `bcHurlbert` and its effective number of species by `HurlbertD` or `bcHurlbertD`.

```
R> Hurlbert(Ps = Paracou618.MC$Ps, k = 2)

[1] 1.985449

R> bcHurlbert(Ns = Paracou618.MC$Ns, k = 2)

[1] 1.986326

R> bcHurlbertD(Ns = Paracou618.MC$Ns, k = 2)

[1] 73.13163
```

Hurlbert's diversity of order 2 is identical to Simpson's diversity.

3.2. Meta-community functions

Meta-community functions allow partitioning diversity according to Patil and Taillie's concept of diversity of a mixture (**?**), *i.e.* α entropy of a meta-community is defined as the weighted average of community entropy, following **?**:

$$ {}^{q}H_{\alpha} = \sum_{i} w_{i} \, {}^{q}_{i}H_{\alpha} \tag{9} $$

${}^{q}_{i}H_{\alpha}$ is the entropy of community i:

$$ {}^{q}_{i}H_{\alpha} = \frac{1 - \sum_{s} p^{q}_{s,i}}{q - 1} = - \sum_{s} p^{q}_{s,i} \ln_{q} p_{s,i} \tag{10} $$

Jost's (**?**) definition of α entropy is not supported explicitly in the package since it only allows partitioning of equally weighted communities. In this particular case, both definitions are identical.

γ entropy of the meta-community is defined as α entropy of a community. β entropy, the difference between γ and α, is the generalized Jensen-Shannon divergence between the species distribution of the meta-community and those of communities (**?**):

$$ {}^{q}H_{\beta} ={}^{q} H_{\gamma} -{}^{q} H_{\alpha} = \sum_{s} p^{q}_{s,i} \ln_{q} \frac{p_{s,i}}{p_{s}} \tag{11} $$

β entropy should be transformed into diversity, *i.e.* an effective number of communities:

$$ {}^{q}D_{\beta} = e_q^{\frac{{}^{q}H_{\beta}}{1-(q-1){}^{q}H_{\alpha}}} \tag{12} $$

Basic meta-community functions

These values can be estimated by the meta-community functions named `AlphaEntropy`, `AlphaDiversity`, `BetaEntropy`, `BetaDiversity`. They accept a `Metacommunity` and an order of diversity q as arguments, and return an `MCentropy` or `MCdiversity` object which can be summarized and plotted. `GammaEntropy` and `GammaDiversity` return a number. Estimation-bias corrections are applied by default:

```
R> e <- AlphaEntropy(Paracou618.MC, q = 1)
R> summary(e)

Neutral alpha entropy of order 1 of metaCommunity Paracou618.MC
with correction: Best

Entropy of communities:
    P006     P018
4.384539 4.706510
Average entropy of the communities:
[1] 4.522322
```

The Shannon α entropy of the meta-community is 4.52. It is the weighted average entropy of communities.

Diversity Partition of a metacommunity

The `DivPart` function calculates everything at once. Its arguments are the same but bias correction is not applied by default. It can be, using the argument `Biased = FALSE`, and the correction chosen by the argument `Correction`. It returns a `DivPart` object which can be summarized (entropy is not printed by `summary`) and plotted:

```
R> p <- DivPart(q = 1, MC = Paracou618.MC, Biased = FALSE)
R> summary(p)

HCDT diversity partitioning of order 1 of metaCommunity Paracou618.MC
 with correction: Best
Alpha diversity of communities:
     P006      P018
 80.20125 110.66525
Total alpha diversity of the communities:
[1] 92.04908
Beta diversity of the communities:
[1] 1.447498
Gamma diversity of the metacommunity:
[1] 133.2409
```

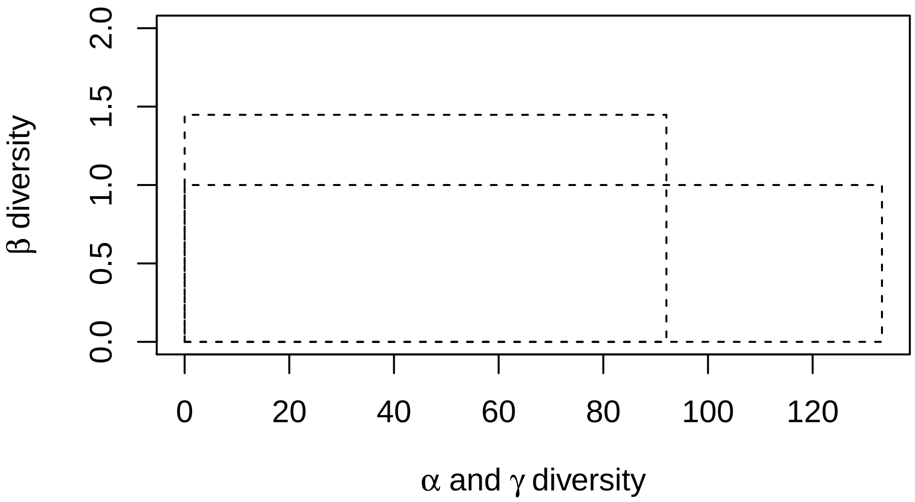

Figure 4. Plot of the diversity partition of the meta-community `Paracou618.MC`. The long rectangle of height 1 represents γ diversity, equal to 133 effective species. The narrower and higher rectangle has the same area: its horizontal size is α diversity (92 effective species) and its height is β diversity (1.45 effective communities).

```
R> p$CommunityAlphaEntropies
```

```
    P006     P018 
4.384539 4.706510 
```

The α diversity of communities is 92 effective species (it is the exponential of the entropy calculated previously). This is more than Simpson diversity (73 species, calculated above) because less frequent species are taken into account. γ diversity of the meta-community is 133 effective species. β diversity is 1.45 effective communities, *i.e.* the two actual communities are as different from each other as 1.45 ones with equal weights and no species in common.

Diversity Estimation of a metacommunity

The `DivEst` function decomposes diversity and estimates confidence interval of α, β and γ diversity following **?**. If the observed species frequencies of a community are assumed to be a realization of a multinomial distribution, they can be drawn again to obtain a distribution of entropy.

```
R> de <- DivEst(q = 1, Paracou618.MC, Biased = FALSE, Correction = "Best",
+      Simulations = 100)
```

```
=====================================================================
```

```
R> summary(de)
```

```
Diversity partitioning of order 1 of MetaCommunity MC
 with correction: Best
Alpha diversity of communities:
     P006      P018 
 80.20125 110.66525 
Total alpha diversity of the communities:
[1] 92.04908
Beta diversity of the communities:
[1] 1.447498
Gamma diversity of the metacommunity:
[1] 133.2409
Quantiles of simulations (alpha, beta and gamma diversity):
      0%       1%     2.5%       5%      10%      25%      50% 
85.01019 85.96966 86.25811 86.99451 87.68495 89.11180 92.11585 
     75%      90%      95%    97.5%      99%     100% 
94.36106 97.20894 98.68814 99.09826 99.76917 99.77629 
      0%       1%     2.5%       5%      10%      25%      50% 
1.393900 1.396654 1.399821 1.410408 1.418374 1.433221 1.449459 
     75%      90%      95%    97.5%      99%     100% 
1.460973 1.475313 1.480472 1.487628 1.499847 1.505753 
      0%       1%     2.5%       5%      10%      25%      50% 
123.4410 124.6241 124.9687 125.5656 127.2835 129.5636 133.2328 
     75%      90%      95%    97.5%      99%     100% 
137.0294 140.0737 141.2884 142.3734 144.2411 145.8396 
```

The result is a `Divest` object which can be summarized and plotted (Figure 5).

The uncertainty of estimation is due to sampling: the distribution of the estimators corresponds to the simulated repetitions of sampling in the original multinomial distribution of species. It ignores the remaining bias of the estimator, which is unknown. Yet, except for $q = 2$, the corrected estimators *are* biased (even though much less than the non-corrected ones), especially when q is small. New estimators to reduce the bias are included in the package regularly.

Diversity Profile of a metacommunity

`DivProfile` calculates diversity profiles, *i.e.* the value of diversity against its order (Figure 6). The result is a `DivProfile` object which can be summarized and plotted.

```
R> dp <- DivProfile(seq(0, 2, 0.2), Paracou618.MC, Biased = FALSE)
R> summary(dp)
```

```
Diversity profile of MetaCommunity MC
 with correction: Best
Diversity against its order:
      Order Alpha Diversity Beta Diversity Gamma Diversity
 [1,]   0.0       238.36351       1.321238       314.93484
 [2,]   0.2       204.17972       1.322729       270.07441
```

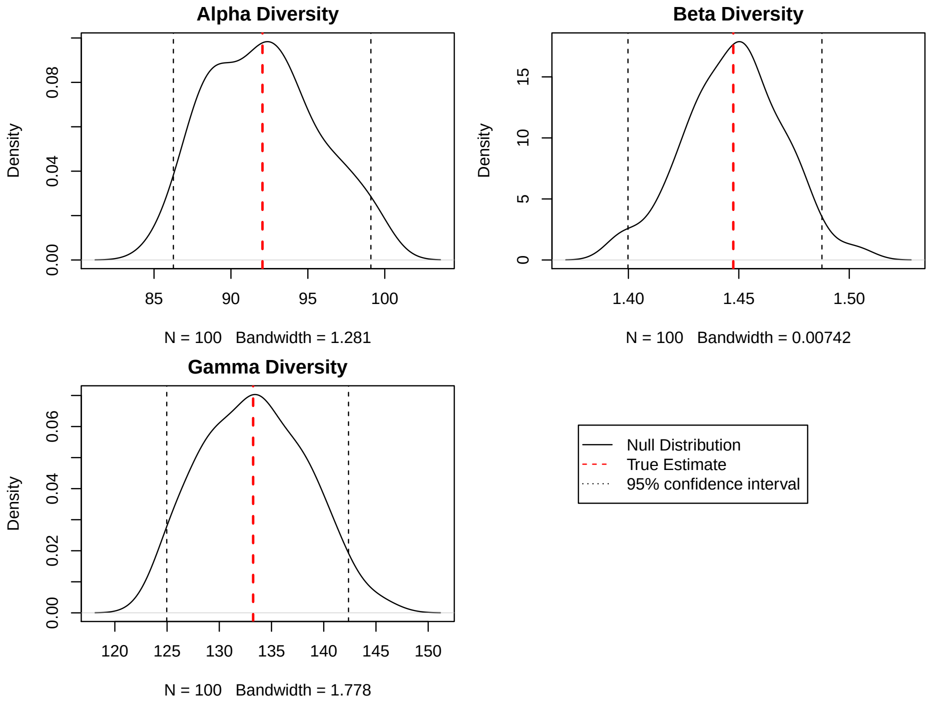

Figure 5. Plot of the diversity estimation of the meta-community `Paracou618.MC`. α, β and γ diversity probability densities are plotted, with a 95% confidence interval.

```
 [3,]   0.4      170.72117      1.334724      227.86565
 [4,]   0.6      139.89024      1.360036      190.25572
 [5,]   0.8      113.37058      1.398840      158.58725
 [6,]   1.0       92.04908      1.447498      133.24089
 [7,]   1.2       75.81819      1.499692      113.70393
 [8,]   1.4       63.87372      1.549289       98.95886
 [9,]   1.6       55.18979      1.592530       87.89138
[10,]   1.8       48.84485      1.628258       79.53203
[11,]   2.0       44.13559      1.656977       73.13163
```

Small orders of diversity give more weight to rare species. P018 can be considered more diverse than P006 because their profiles (Figure 6, top right) do not cross (?): its diversity is systematically higher. The shape of the β diversity profile shows that the communities are more diverse when their dominant species are considered.

Alternative functions

Beta entropy can also be calculated by a set of functions named after the community functions, such as `TsallisBeta`, `bcTsallisBeta`, `SimpsonBeta`, etc. which require two vectors of abundances or probabilities instead of a `MetaCommunity` object: that of the community and the expected one (usually that of the meta-community). Bias correction is currently limited to Chao and Shen's correction. The example below calculates the Shannon β entropy of the first community of Paracou618 and the meta-community.

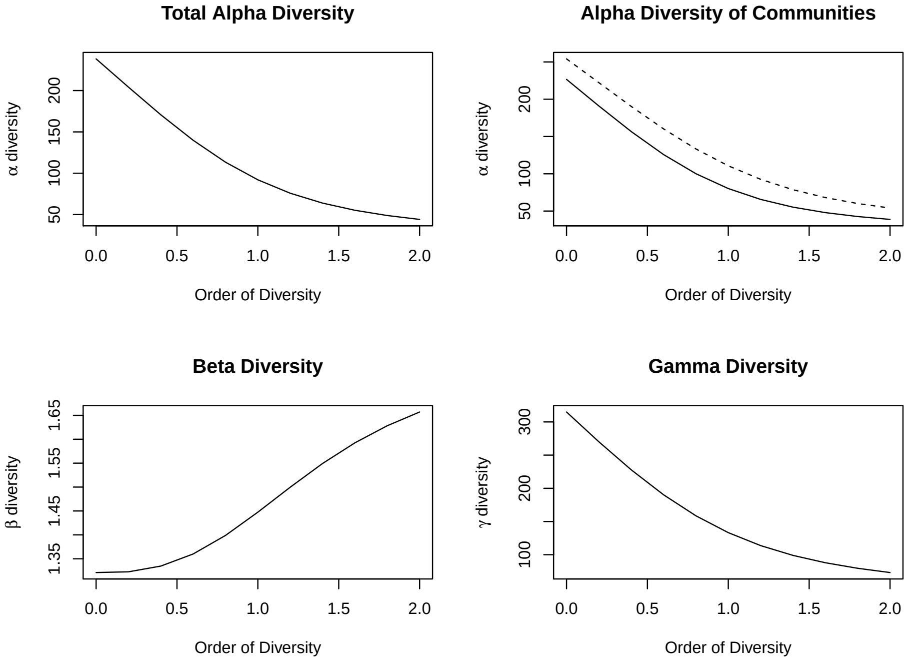

Figure 6. Diversity profile of the meta-community `Paracou618.MC`. Values are the number of effective species (α and γ diversity) and the effective number of communities (β diversity). Community P006 is represented by the solid line and community P018 by the dotted line. α and γ diversity decrease from $q = 0$ (number of species) to $q = 2$ (Simpson diversity) by construction.

```
R> ShannonBeta(Paracou618.MC$Psi[, 1], Paracou618.MC$Ps)

[1] 0.3499358
```

These functions are available for particular uses, when a `MetaCommunity` is not available or not convenient to use (*e.g.* simulations). Meta-community functions are preferred in general.

4. Phylogenetic diversity

Phylogenetic or functional diversity generalizes HCDT diversity, considering the distance between species (**?**). Here, all species take place in an ultrametric phylogenetic or functional tree (Figure 7). The tree is cut into slices, delimited by two nodes. The first slice starts at the bottom of the tree and ends at the first node. In slice k, L_k leaves are found. The probabilities of occurrence of the species belonging to branches that were below leaf l in the original tree are summed to give the grouped probability $u_{k,l}$. HCDT entropy can be calculated in slice k:

$$ {}^q_kH = -\sum_l u^q_{k,l} \ln_q u_{k,l} \tag{13}$$

Then, it is summed over the tree slices. Phyloentropy can be normalized or not. We normalize

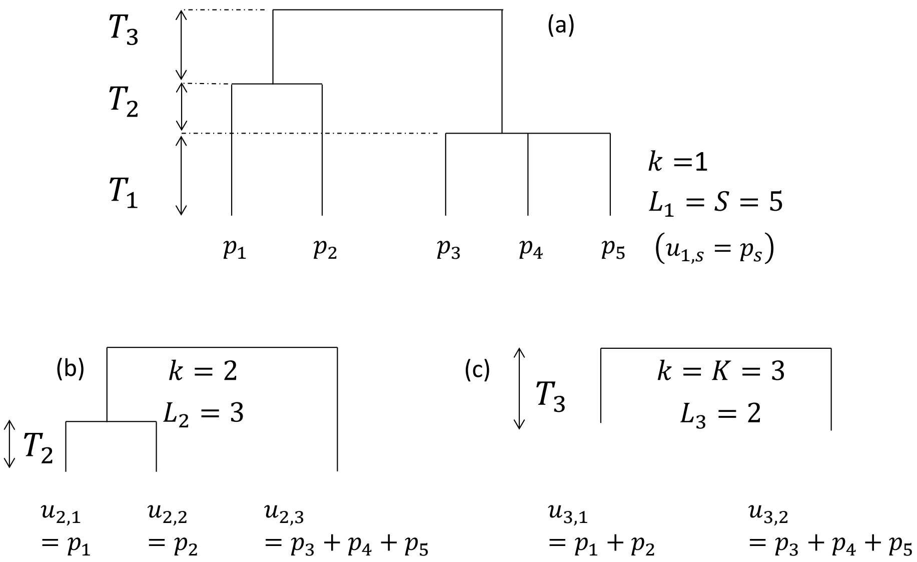

Figure 7. Hypothetical ultrametric tree. (a) The whole tree contains three slices, delimited by two nodes. The length of slices is T_k. (b) Focus on slice 2. The tree without slice 1 is reduced to 3 leaves. Frequencies of collapsed species are $u_{k,l}$. (c) Slice 3 only.

it so that it does not depend on the tree height:

$$^{q}\overline{H}\left(T\right) = \sum_{k=1}^{K} \frac{T_k}{T} {}^{q}_{k}H \tag{14}$$

Unnormalized values are multiplied by the tree height, such as $^{q}PD(T)$ (**?**).

Phyloentropy is calculated as HCDT entropy along the slices of the trees applying possible estimation-bias corrections, summed, possibly normalized, and finally transformed into diversity:

$$^{q}\overline{D}\left(T\right) = e_q^{{}^{q}\overline{H}(T)} \tag{15}$$

4.1. Community functions

`PhyloEntropy` and the estimation-bias-corrected `bcPhyloEntropy` are the phylogenetic analogs of `Tsallis` and `bcTsallis`. They accept the same arguments plus an ultrametric tree of class `hclust` or `phylog`, and `Normalize`, a boolean to normalize the tree height to 1 (by default).

Phylogenetic diversity is calculated by `PhyloDiversity` or `bcPhyloDiversity`, analogous to the species-neutral diversity functions `Diversity` and `bcDiversity`.

Results are either a `PhyloDiversity` or a `PhyloEntropy` object, which can be plotted (Figure 8) and summarized.

```
R> phd <- bcPhyloDiversity(Paracou618.MC$Ns, q = 1, Tree = Paracou618.Taxonomy,
+      Normalize = TRUE)
R> summary(phd)
```

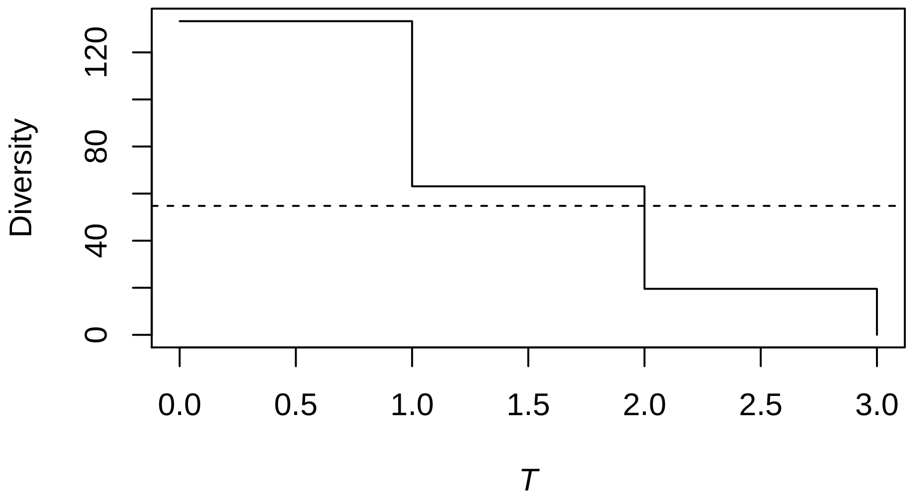

Figure 8. Plot of the γ phylodiversity estimation of the meta-community `Paracou618.MC`. The effective number of taxa of Shannon diversity is plotted against the distance from the leaves of the phylogenetic tree. Here, the tree is based on a rough taxonomy, so diversity of species, genera and families are the three levels of the curve. The dotted line represents the value of phylodiversity.

```
alpha or gamma phylogenetic or functional diversity of order 1
of distribution Paracou618.MC$Ns
 with correction: Best
Phylogenetic or functional diversity was calculated according to the tree
Paracou618.Taxonomy

Diversity is  normalized

Diversity equals: 54.78324
```

The phylogenetic diversity of order 1 of the Paracou dataset is 55 effective species: 55 totally different species (only connected by the root of the tree) with equal probabilities would have the same entropy. It can be compared to its species-neutral diversity, 133 species. The latter is the diversity of the first slice of the tree. When going up the tree, diversity decreases because species collapse. On Figure 8, diversity of the second slice, between $T = 1$ and $T = 2$, is that of genera (63 effective genera) and the last slice contains (20 effective families). The phylogenetic entropy of the community is the average of the entropy along slices, weighted by the slice lengths. Diversity can not be averaged the same way.

A less trivial phylogeny would contain many slices, resulting in as many diversity levels with respect to T.

The `AllenH` function is close to `PhyloEntropy`: it also calculates phyloentropy but the algorithm is that of **?** for $q = 1$ and that of **?** for $q \neq 1$. It is much faster since it does not require calculating entropy for each slice of the tree but it does not allow estimation-bias correction. `ChaoPD` calculates phylodiversity according to **?**, with the same advantages and limits compared to PhyloDiversity.

For convenience, PDFD and Rao functions are provided to calculate unnormalized phyloentropy of order 0 and 2.

4.2. Meta-community functions

DivPart, DivEst and DivProfile functions return phylogenetic entropy and diversity values instead of species-neutral ones if a tree is provided in the arguments.

```
R> dp <- DivPart(q = 1, Paracou618.MC, Biased = FALSE, Correction = "Best",
+      Tree = Paracou618.Taxonomy)
R> summary(dp)

HCDT diversity partitioning of order 1 of metaCommunity Paracou618.MC
 with correction: Best
Phylogenetic or functional diversity was calculated
according to the tree
Paracou618.Taxonomy

Diversity is  normalized

Alpha diversity of communities:
    P006     P018
36.66108 51.98951
Total alpha diversity of the communities:
[1] 42.57235
Beta diversity of the communities:
[1] 1.286827
Gamma diversity of the metacommunity:
[1] 54.78324
```

The decomposition is interpreted as the species-neutral one: γ diversity is 55 effective species, made of 1.3 effective communities of 43 effective species.

Other meta-community functions, such as AlphaEntropy behave the same way:

```
R> summary(BetaEntropy(Paracou618.MC, q = 2, Tree = Paracou618.Taxonomy,
+      Correction = "None", Normalize = FALSE))

HCDT beta entropy of order 2 of metaCommunity Paracou618.MC
with correction: None

Phylogenetic or functional entropy was calculated according to the tree
Paracou618.Taxonomy

Entropy is not normalized
Entropy of communities:
      P006       P018 
0.04117053 0.02325883 
Average entropy of the communities:
[1] 0.03350547
```

Compare with Rao's `divc` computed by **ade4**:

```
R> library("ade4")
R> divc(as.data.frame(Paracou618.MC$Wi), disc(as.data.frame(Paracou618.MC$Nsi),
+      Paracou618.Taxonomy$Wdist))
```

```
                  diversity
Paracou618.MC$Wi 0.03350547
```

5. Similarity-based diversity

? introduced similarity-based diversity of a community ${}^{q}D^{Z}$. A matrix $\mathbf{Z}$ describes the similarity between pairs of species, defined between 0 and 1. A species ordinariness is its average similarity with all species (weighted by species frequencies), including similarity with itself (equal to 1). Similarity-based diversity is the reciprocal of the generalized average of order q (**?**) of the community species ordinariness.

The `Dqz` function calculates similarity-based diversity. Its arguments are the vector of probabilities of occurrences of the species, the order of diversity and the similarity matrix $\mathbf{Z}$. The `bcDqz` function allows estimation-bias correction.

This example calculates the γ diversity of the meta-community Paracou. First, the similarity matrix is calculated from the distance matrix between all pairs of species as 1 minus normalized dissimilarity.

```
R> DistanceMatrix <- as.matrix(Paracou618.dist)
R> Z <- 1 - DistanceMatrix/max(DistanceMatrix)
R> bcDqz(Paracou618.MC$Ns, q = 2, Z)
```

```
[1] 1.483027
```

If $\mathbf{Z}$ is the identity matrix, similarity-based diversity equals HCDT diversity:

```
R> Dqz(Paracou618.MC$Ps, q = 2, Z = diag(length(Paracou618.MC$Ps)))
```

```
[1] 68.7215
```

```
R> Diversity(Paracou618.MC$Ps, q = 2)
```

```
[1] 68.7215
```

Functional diversity of order 2 is only 1.48 effective species, which is very small compared to 69 effective species for Simpson diversity. 1.48 equally-frequent species with similarity equal to 0 would have the same functional diversity as the actual community (made of 425 species). This means that species are very similar from a functional point of view. The very low values returned by ${}^{q}D^{Z}$ are questioned by **?** and discussed in depth by **?**: the choice of the similarity matrix is not trivial.

The similarity-based entropy of a community ${}^{q}H^{Z}$ (**??**) has the same relations with diversity as HCDT entropy and Hill numbers. The `Hqz` function calculates it:

```
R> Hqz(Paracou618.MC$Ps, q = 2, Z)

[1] 0.3208152

R> lnq(Dqz(Paracou618.MC$Ps, q = 2, Z), q = 2)

[1] 0.3208152
```

As species-neutral entropy, ${}^{q}H^{Z}$ has no straightforward interpretation beyond the average surprise of a community.

All meta-community functions can be used to estimate similarity-based diversity: argument `Z` must be provided:

```
R> e <- AlphaEntropy(Paracou618.MC, q = 1, Z = Z)
R> summary(e)

Similarity-based alpha entropy of order 1 of metaCommunity
Paracou618.MC with correction: Best

Phylogenetic or functional entropy was calculated according to the similarity matrix
Z

Entropy of communities:
     P006      P018
0.3945541 0.3934725
Average entropy of the communities:
[1] 0.3940912
```

The α functional entropy of the meta-community is the average entropy of communities.

6. Advanced tools

The package comes with a set of tools to realize frequents tasks: run Monte-Carlo simulations on a community, quickly calculate its diversity profile, apply a function to a species distribution along a tree, and manipulate meta-communities.

6.1. Random communities

The rCommunity function allows creating random communities. Their species probability distribution can be drawn in a well-known distribution (such as a log-normal one) or obtained from the data, just by dividing abundances by the total number of individuals (?), or derived from a more sophisticated model by ?. Finally, the specified number of communities are drawn in a multinomial distribution.

The log-normal (?), the log-series (?), the geometric (?), and the broken-stick (?) distributions can be simulated.

This example code draws a single community of 1000 individuals according to a log-normal distribution with 300 species. Many species are not observed in the 1000-individual sample: the observed number of species is shown, with an estimation of the actual number (which should be 300). The simulated community is plotted, with its log-normal distribution fitted from the data (Figure 9). Estimated parameters can be compared to the original ones.

```
R> NsRef <- rCommunity(n = 1, size = 1000, S = 300, Distribution = "lnorm",
+      sd = 1)
R> Richness(as.ProbaVector(NsRef))

[1] 229

R> bcRichness(NsRef)

[1] 279.8337

R> plot(NsRef, Distribution = "lnorm")

$mu
[1] 1.071834

$sigma
[1] 0.8735653
```

6.2. Entropy of Monte-Carlo simulated communities

The EntropyCI function is a versatile tool to simplify simulations. Simulated communities are obtained by random draws in a multinomial distribution of species and their entropy is calculated. The arguments of EntropyCI are an entropy function (any entropy function of the package accepting a vector of species abundances, such as bcTsallis), the number of simulations to run, the observed species frequencies and the method to obtain probabilities for the multinomial distribution (the same as that of rCommunity).

The result is a numeric vector containing the entropy value of each simulated community. Entropy can be finally transformed into diversity (but it is not correct to use a diversity function in simulations because the average simulated value must be calculate and only entropy can be averaged).

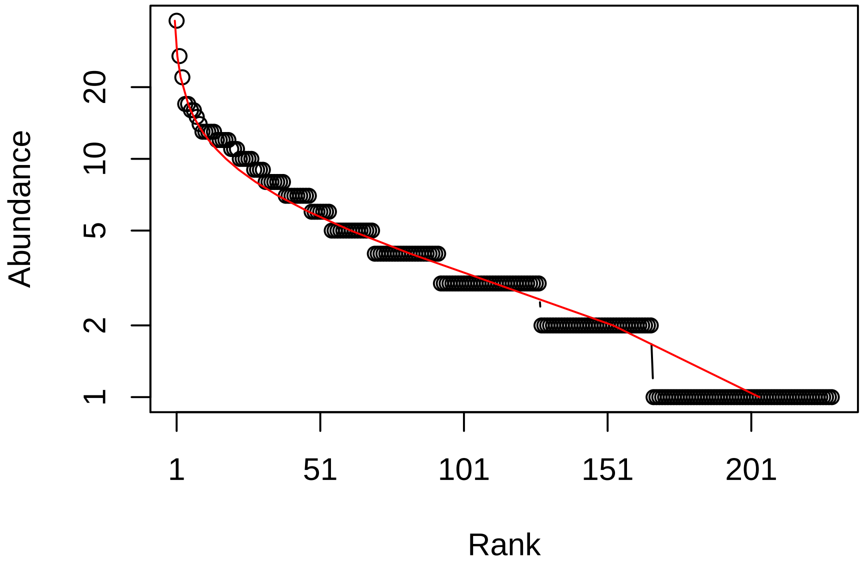

Figure 9. Whittaker plot of a simulated community from a log-normal distrbution. The red line is the fit of a log-normal distribution estimated from the data.

This example shows how to use the function. First, the distribution of the γ HCDT entropy of order 1 (Shannon entropy) of the Paracou meta-community is calculated and transformed into diversity. Then, the actual diversity is calculated and completed by the 95% confidence interval of the simulated values.

```
R> SimulatedDiversity <- expq(EntropyCI(FUN = Tsallis, Simulations = 100,
+      Ns = PAbd, q = 1), q = 1)

======================================================================

R> Diversity(PAbd, q = 1)

[1] 133.2409

R> quantile(SimulatedDiversity, probs = c(0.025, 0.975))

    2.5%    97.5%
121.7348 143.1577
```

These results are identical to those of the `DivEst` function but a single community can be addressed (`DivEst` requires a `MetaCommunity`).

6.3. Diversity or Entropy Profile of a community

This function is used to calculate diversity or entropy profiles based on community functions such as `Tsallis` or `ChaoPD`. It is similar to `DivProfile` but does not require a `Metacommunity` for argument. It can compute a bootstrap confidence envelope of the estimation of the profile, like `EntropyCI`. It returns a `CommunityProfile` object which can be plotted. Profiles can be added to an existing plot by the `CEnvelope` function.

This example evaluates bias correction on the diversity profile of the Paracou dataset. First, diversity profiles are calculated with and without bias correction. The corrected profile is calculated with its confidence envelope:

```
R> bcProfile <- CommunityProfile(Diversity, PAbd, NumberOfSimulations = 10)

======================================================================

R> Profile <- CommunityProfile(Diversity, PProba)
```

Then, they can be plotted altogether (Figure 10):

```
R> plot(bcProfile)
R> CEnvelope(Profile, lty = 3)
R> legend("topright", c("Bias Corrected", "Biased"), lty = c(1,
+      3), inset = 0.02)
```

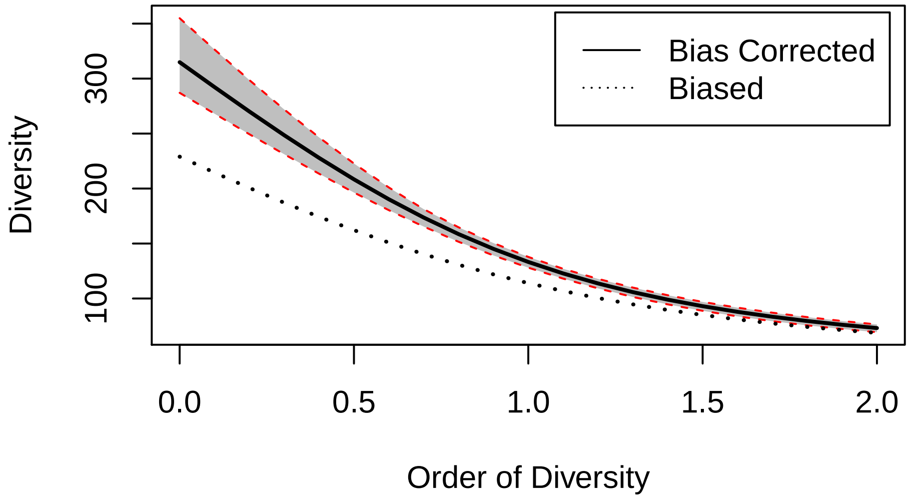

Figure 10. γ diversity profile of the the meta-community `Paracou618.MC`, without bias correction (dotted line) and with correction (solid line). .

6.4. Applying a Function over a Phylogenetic Tree

The `PhyloApply` function is used to apply an entropy community function (generally `bcTsallis`) along a tree, the same way `lapply` works with a list.

This example shows how to calculate Shannon entropy along the tree containing the taxonomy to obtain species, genus and family entropy shown on figure 8:

```
R> pa <- PhyloApply(Tree = Paracou618.Taxonomy, FUN = bcTsallis,
+      NorP = Paracou618.MC$Ns)
R> summary(pa)

bcTsallis applied to Paracou618.MC$Ns along the tree
Paracou618.Taxonomy

Results are  normalized

The average value is: 4.003384

Values along the tree are:
       1        2        3
4.892159 4.144688 2.973306

R> exp(pa$Cuts)

        1         2         3
133.24089  63.09794  19.55647

R> exp(pa$Total)
```

```
[1] 54.78324
```

6.5. Manipulation of meta-communities

Several meta-communities, combined in a list, can be merged two different ways: the `MergeMC` function simplifies hierarchical partitioning of diversity: it creates a new meta-community whose communities are the original meta-communities aggregated data. The α entropy of the new meta-community is the weighted average γ entropy of the original meta-communities.

`MergeC` combines the communities of several meta-communities to create a single meta-community containing them all. Last, `ShuffleMC` randomly shuffles communities accross meta-communities to allow simulations to test differences between meta-communities.

This example shows how to do this. A first meta-community is created, weights of communities are proportional to their number of individuals:

```
R> (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")))
```

```
    C1 C2 C3
sp1 10  0 25
sp2 10 20 15
sp3 10 35  0
sp4 10  5  2
```

```
R> w <- colSums(df)
R> MC1 <- MetaCommunity(Abundances = df, Weights = w)
```

Then a second one:

```
R> (df <- data.frame(C1 = c(10, 4), C2 = c(3, 4), row.names = c("sp1",
+      "sp5")))
```

```
    C1 C2
sp1 10  3
sp5  4  4
```

```
R> w <- colSums(df)
R> MC2 <- MetaCommunity(Abundances = df, Weights = w)
```

They can be merged to obtain a single meta-community containing all original communities:

```
R> mergedMC1 <- MergeC(list(MC1, MC2))
R> mergedMC1$Nsi
```

```
    MC1.C1 MC1.C2 MC1.C3 MC2.C1 MC2.C2
sp1     10      0     25     10      3
sp2     10     20     15      0      0
sp3     10     35      0      0      0
sp4     10      5      2      0      0
sp5      0      0      0      4      4
```

They can also be merged considering each of them as a community of a higher-level meta-community:

```
R> mergedMC2 <- MergeMC(list(MC1, MC2), Weights = sapply(list(MC1,
+     MC2), function(x) (x$N)))
R> mergedMC2$Nsi
```

```
    MC1 MC2
sp1  35  13
sp2  45   0
sp3  45   0
sp4  17   0
sp5   0   8
```

Hierarchical diversity partitioning can then be achieved:

```
R> dpAll <- DivPart(q = 1, MC = mergedMC2)
R> summary(dpAll)
```

```
HCDT diversity partitioning of order 1 of metaCommunity mergedMC2

Alpha diversity of communities:
     MC1      MC2 
3.772161 1.943574 
Total alpha diversity of the communities:
[1] 3.463277
Beta diversity of the communities:
[1] 1.236351
Gamma diversity of the metacommunity:
[1] 4.281826
```

The γ diversity of the top assemblage (MC1 and MC2) is 4.28 effective species, made of 1.24 effective meta-communities of 3.46 effective species. The α diversity of each meta-community of the top assemblage is their γ diversity when it is partitioned in turn:

```
R> dpMC1 <- DivPart(q = 1, MC = MC1)
R> summary(dpMC1)
```

```
HCDT diversity partitioning of order 1 of metaCommunity MC1
```

```
Alpha diversity of communities:
      C1       C2       C3
4.000000 2.429521 2.273918
Total alpha diversity of the communities:
[1] 2.741671
Beta diversity of the communities:
[1] 1.375862
Gamma diversity of the metacommunity:
[1] 3.772161
```

The γ diversity of MC1 is 3.77 effective species, made of 1.38 effective meta-communities of 2.74 effective species. The same decomposition can be done for MC2.

7. Conclusion

The entropart package allows estimating biodiversity according to the framework based on HCDT entropy, the correction of its estimation-bias (**??**) and its transformation into equivalent numbers of species (**???**). Phylogenetic or functional diversity (**?**) can be estimated, considering phyloentropy as the average species-neutral diversity over slices of a phylogenetic or functional tree (**?**). Similarity-based diversity **?** can be used to estimate (**?**) functional diversity from a similarity or dissimilarity matrix between species without requiring building a dendrogram and thus preserving the topology of species (**??**).

The classical diversity estimators (Shannon and Simpson entropy) can be found in many R packages. **vegetarian** (**?**) allows calculating Hill numbers and partitioning them according to Jost's framework. Bias correction is never available except in the **EntropyEstimation** (**?**) package which provides the Zhang and Grabchak's estimators of entropy and diversity and their asymptotic variance (not included in **entropart**). Phylodiversity and similarity-based diversity are not available in any package as far as we know. So we believe **entropart** is a useful toolbox for ecologists who need to estimate the diversity of actual, undersampled communities and to partition it.

Affiliation:

Eric Marcon
AgroParisTech
Campus agronomique, BP 316
97310 Kourou, French Guiana
E-mail: eric.marcon@ecofog.gf

Bruno Hérault
Cirad
Campus agronomique, BP 316
97310 Kourou, French Guiana
E-mail: bruno.herault@ecofog.gf