A Quick Introduction to iNEXT.seq via Example

Anne Chao

2024-07-20

`iNEXT.seq` (iNterpolation and EXTrapolation for phylogenetic beta diversity and dissimilarity measure for genetic sequence data) is an R package. In this document, we provide a quick introduction demonstrating how to run `iNEXT.seq`. Detailed information about `iNEXT.seq` functions is provided in the iNEXT.seq manual, also available in CRAN. An online version of iNEXT.seq-online is also available for users without an R background.

`iNEXT.seq` introduces a novel method to measure UniFrac distance by applying dissimilarity measures based on Chiu's et al.'s multiple-assemblage decomposition (Chiu, Jost, and Chao (2014)). It primarily considers three measures of Hill numbers: species richness (`q = 0`), Shannon diversity (`q = 1`, the exponential of Shannon entropy), and Simpson diversity (`q = 2`, the inverse of Simpson concentration).

Utilizing the observed sample of OTU count data (the "reference sample"), `iNEXT.seq` calculates UniFrac distance estimates and associated confidence intervals for coverage-based rarefaction and extrapolation (R/E) curves. By performing a monotonic transformation on phylogenetic beta diversity for rarefied and extrapolated samples, based on standardized levels of sample completeness (measured by sample coverage), `iNEXT.seq` plots diversity estimates against sample coverage on a coverage-based sampling curve. In addition, `iNEXT.seq` provides asymptotic estimates phylogenetic diversity and dissimilarity for multiple-assemblage and hierarchical decompositions under Chiu's et al.'s (2014) and Routledge's (1979) framework, as well as diversity profiles for order q.

`iNEXT.seq` features two statistical analyses for multiple assemblages (non-asymptotic and asymptotic), and asymptotic statistical analysis for hierarchical structure data:

1. A non-asymptotic approach based on interpolation and extrapolation for multiple-assemblage phylogenetic diversity

Two types of R/E sampling curves:

- **Sample-size-based (or size-based) R/E curves**: This type of sampling curve plots standardized phylogenetic gamma and alpha diversity with respect to sample size.
- **Coverage-based R/E sampling curves**: This type of sampling curve plots standardized phylogenetic gamma, alpha, and beta diversity as well as four classes of dissimilarity measures with respect to sample coverage (an objective measure of sample completeness).

2. An asymptotic approach to infer asymptotic phylogenetic diversity for multiple-assemblage data

iNEXT.seq computes the estimated asymptotic phylogenetic gamma, alpha, and beta diversity as well as four classes of dissimilarity measures and also plots profiles (q-profiles) for order q between 0 and 2 (by default), in comparison with the observed diversity and dissimilarity measures. Additionally, two types of beta diversity decomposition calculations are provided:

- **Relative**: Routledge's (1979) approach for comparing node relative abundances for size-weighted assemblages.
- **Absolute**: Chiu et al.'s (2014) approach for comparing node raw abundances.

3. An asymptotic approach to infer asymptotic phylogenetic diversity for hierarchical structure data

Based on above asymptotic approach for multiple assemblages, we introduce a hierarchical structural estimation, calculate the hierarchical phylogenetic diversity and dissimilarity measures, and also plots profiles (q-profiles) for order q.

Note that sufficient data are needed to run iNEXT.seq. If your data comprises only a few species and their phylogenies, it is probable that the data lack sufficient information to run iNEXT.seq.

HOW TO CITE iNEXT.seq

If you publish your work based on the results from the `iNEXT.seq` package, you should make references to the following methodology paper:

- Chiu, C.-H., Jost, L. and Chao*, A. (2014). Phylogenetic beta diversity, similarity, and differentiation measures based on Hill numbers. *Ecological Monographs*, 84, 21-44.
- Routledge, R. (1979). Diversity indices: which ones are admissible? *Journal of Theoretical Biology*, 76(4), 503-515.

SOFTWARE NEEDED TO RUN iNEXT.seq IN R

- Required: R
- Suggested: RStudio IDE

HOW TO RUN iNEXT.seq:

The `iNEXT.seq` package is available from CRAN and can be downloaded with a standard R installation procedure or can be downloaded from Anne Chao's iNEXT.seq_github using the following commands. For a first-time installation, an additional visualization extension package (`ggplot2`) must be installed and loaded.

```
## install iNEXT.seq package from CRAN
install.packages("iNEXT.seq")
## install the latest version from github
install.packages('devtools')
library(devtools)
install_github('ChunYuYeh/iNEXT.seq')
## import packages
library(iNEXT.seq)
```

Here are six main functions we provide in this package :

Two functions for non-asymptotic analysis for multiple-assemblage phylogenetic diversity with graphical displays:

- **iNEXTseq** : Computes standardized phylogenetic diversity estimates of order q = 0, 1 and 2 for rarefied and extrapolated samples at specified sample coverage values and sample sizes.
- **ggiNEXTseq** : Visualizing the output from the function `iNEXTseq`

Two functions for asymptotic analysis for multiple-assemblage phylogenetic diversity with graphical displays:

- **ObsAsyPD** : Computes observed and asymptotic diversity of order q between 0 and 2 (in increments of 0.2) for multiple-assemblage phylogenetic diversity.
- **ggObsAsyPD** : Visualizing the output from the function `ObsAsyPD`

Two functions for asymptotic analysis for hierarchical phylogenetic diversity with graphical displays:

- **hierPD** : Computes observed and asymptotic diversity of order q between 0 and 2 (in increments of 0.2) for hierarchical phylogenetic diversity.
- **gghierPD** : Visualizing the output from the function `hierPD`

ANALYSIS FOR MULTIPLE ASSEMBLAGES

DATA INPUT FORMAT

Individual-based OTU count data

Input data for each data set with several assemblages/sites include samples species abundances in an empirical sample of n individuals ("reference sample"). When there are N assemblages in a data set, input data consist of a list with an S by N abundance matrix; For M data sets consisting N assemblages, input data should be M lists of S by N abundance matrix.

A data set (a small example dataset from a human esophageal community) is included in `iNEXT.seq` package for illustration. The data consist a list with four species-by-assemblage data.frames ("BC", "BD", "CD" and "BCD"). For the data, the following commands display how to compute estimate at specified sample coverage.

Run the following code to view first list of `esophagus` OTU count data: (Here we only show the first ten rows for the matrix)

```
data("esophagus")
esophagus[1]
```

```
#> $esophagus_BC
#>               B  C
#> OTU_59_8_22 50 19
#> OTU_59_5_13  0  2
#> OTU_59_8_12  2 13
#> OTU_65_3_22  0  2
#> OTU_65_5_1   0  0
#> OTU_65_1_10  0  0
#> OTU_65_7_12  3  2
#> OTU_59_6_1   0  1
#> OTU_65_2_17  6  1
#> OTU_65_9_26  0  0
```

Phylogenetic tree for phylogenetic diversity

To perform phylogenetic diversity analysis, the phylogenetic tree (in Newick format) spanned by species observed in the pooled data is required. For the data set `esophagus`, the phylogenetic tree for all observed species (including species in “B”, “C” and “D”) is stored in the file `esophagus_tree`. A partial list of the tip labels and node labels (not required) are shown below.

```
data("esophagus_tree")
esophagus_tree
```

```
#> 
#> Phylogenetic tree with 58 tips and 57 internal nodes.
#> 
#> Tip labels:
#>   OTU_59_8_22, OTU_59_5_13, OTU_59_8_12, OTU_65_3_22, OTU_65_5_1, OTU_65_1_10, ...
#> 
#> Rooted; includes branch lengths.
```

MAIN FUNCTION: iNEXTseq()

We first describe the main function `iNEXTseq()` with default arguments:

```
iNEXTseq(data, q = c(0, 1, 2), base = "coverage", level = NULL, nboot = 10,
         conf = 0.95, PDtree = NULL, PDreftime = NULL)
```

The arguments of this function are briefly described below, and will be explained in more details by illustrative examples in later text. This main function computes gamma, alpha and beta diversity estimates of order q at specified sample coverage or sample size. By default of `base = “size”` and `level = NULL`, then this function computes the gamma and alpha diversity estimates up to double the reference sample size in each region. If `base = “coverage”` and `level = NULL`, then this function computes the gamma, alpha, beta diversity, and four dissimilarity-turnover indices estimates up to one (for q = 1, 2) or up to the coverage of double the reference sample size (for q = 0).

Argument	Description
`data`	OTU count data can be input as a `matrix/data.frame` (species by assemblages), or a list of `matrices/data.frames`, each matrix represents species-by-assemblages abundance matrix.
`q`	a numerical vector specifying the diversity orders. Default is `c(0, 1, 2)`.
`base`	Sample-sized-based rarefaction and extrapolation for gamma and alpha diversity (`base = “size”`) or coverage-based rarefaction and extrapolation for gamma, alpha and beta diversity (`base = “coverage”`). Default is `base = “coverage”`.
`level`	A numerical vector specifying the particular value of sample coverage (between 0 and 1 when `base = “coverage”`) or sample size (`base = “size”`). `level = 1` (`base = “coverage”`) means complete coverage (the corresponding diversity represents asymptotic diversity). If `base = “size”` and `level = NULL`, then this function computes the gamma and alpha diversity estimates up to double the reference sample size. If `base = “coverage”` and `level = NULL`, then this function computes the gamma and alpha diversity estimates up to one (for `q = 1, 2`) or up to the coverage of double the reference sample size (for `q = 0`); the corresponding beta diversity and dissimilarity are computed up to the same maximum coverage as the alpha diversity.

Argument	Description
nboot	a positive integer specifying the number of bootstrap replications when assessing sampling uncertainty and constructing confidence intervals. Bootstrap replications are generally time consuming. Enter `0` to skip the bootstrap procedures. Default is `10`. Note that large bootstrap replication needs more run time.
conf	a positive number < 1 specifying the level of confidence interval. Default is `0.95`.
PDtree	a `phylo`, a phylogenetic tree in Newick format for all observed species in the pooled assemblage.
PDreftime	a numerical value specifying reference time for PD. Default is `NULL` (i.e., the age of the root of PDtree).

This function returns an `"iNEXTseq"` object which can be further used to make plots using the function `ggiNEXTseq()` to be described below.

When `base = 'coverage'`, the `iNEXTseq()` function returns the `"iNEXTseq"` object including seven data frames for each data sets:

- gamma
- alpha
- beta
- 1-C (Sorensen-type non-overlap)
- 1-U (Jaccard-type non-overlap)
- 1-V (Sorensen-type turnover)
- 1-S (Jaccard-type turnover)

When `base = 'size'`, the `iNEXTseq()` function returns the `"iNEXTseq"` object including two data frames for each data sets:

- gamma
- alpha

Rarefaction/Extrapolation Via Examples

Run the `iNEXTseq()` function with `esophagus` data to compute multiple-assemblage phylogenetic diversity standardized by sample coverage. (Here we only show the first six rows for each output data frame)

```
data("esophagus")
data("esophagus_tree")

out = iNEXTseq(data = esophagus[1], q = c(0, 1, 2), nboot = 10,
               PDtree = esophagus_tree, PDreftime = NULL)
```

```
#> $gamma
#>         Dataset Order.q    SC   Size Gamma      Method  s.e.   LCL   UCL Diversity Reftime
#> 1 esophagus_BC       0 0.500  7.099 2.850 Rarefaction 0.138 2.580 3.121        PD       1
#> 2 esophagus_BC       0 0.525  7.823 2.989 Rarefaction 0.148 2.698 3.279        PD       1
#> 3 esophagus_BC       0 0.550  8.639 3.135 Rarefaction 0.159 2.822 3.447        PD       1
#> 4 esophagus_BC       0 0.575  9.565 3.290 Rarefaction 0.172 2.953 3.628        PD       1
#> 5 esophagus_BC       0 0.600 10.637 3.458 Rarefaction 0.187 3.092 3.824        PD       1
#> 6 esophagus_BC       0 0.625 11.899 3.642 Rarefaction 0.203 3.245 4.039        PD       1
#> 
#> $alpha
#>         Dataset Order.q    SC   Size Alpha      Method  s.e.   LCL   UCL Diversity Reftime
#> 1 esophagus_BC       0 0.500 13.452 2.677 Rarefaction 0.136 2.410 2.944        PD       1
#> 2 esophagus_BC       0 0.525 14.783 2.802 Rarefaction 0.145 2.519 3.086        PD       1
#> 3 esophagus_BC       0 0.550 16.277 2.934 Rarefaction 0.153 2.633 3.235        PD       1
#> 4 esophagus_BC       0 0.575 17.962 3.073 Rarefaction 0.163 2.753 3.393        PD       1
#> 5 esophagus_BC       0 0.600 19.904 3.222 Rarefaction 0.174 2.881 3.563        PD       1
#> 6 esophagus_BC       0 0.625 22.170 3.384 Rarefaction 0.186 3.019 3.749        PD       1
#> 
#> $beta
#>         Dataset Order.q    SC   Size  Beta      Method  s.e.   LCL   UCL Diversity Reftime
#> 1 esophagus_BC       0 0.500 13.452 1.065 Rarefaction 0.018 1.029 1.100        PD       1
#> 2 esophagus_BC       0 0.525 14.783 1.067 Rarefaction 0.019 1.030 1.103        PD       1
#> 3 esophagus_BC       0 0.550 16.277 1.068 Rarefaction 0.019 1.030 1.106        PD       1
#> 4 esophagus_BC       0 0.575 17.962 1.071 Rarefaction 0.020 1.031 1.110        PD       1
#> 5 esophagus_BC       0 0.600 19.904 1.073 Rarefaction 0.021 1.031 1.115        PD       1
#> 6 esophagus_BC       0 0.625 22.170 1.076 Rarefaction 0.022 1.032 1.120        PD       1
#> 
#> $`1-C`
#>         Dataset Order.q    SC   Size Dissimilarity      Method  s.e.   LCL   UCL Diversity Reftime
#> 1 esophagus_BC       0 0.500 13.452         0.065 Rarefaction 0.018 0.029 0.100        PD       1
#> 2 esophagus_BC       0 0.525 14.783         0.067 Rarefaction 0.019 0.030 0.103        PD       1
#> 3 esophagus_BC       0 0.550 16.277         0.068 Rarefaction 0.019 0.030 0.106        PD       1
#> 4 esophagus_BC       0 0.575 17.962         0.071 Rarefaction 0.020 0.031 0.110        PD       1
#> 5 esophagus_BC       0 0.600 19.904         0.073 Rarefaction 0.021 0.031 0.115        PD       1
#> 6 esophagus_BC       0 0.625 22.170         0.076 Rarefaction 0.022 0.032 0.120        PD       1
#> 
#> $`1-U`
#>         Dataset Order.q    SC   Size Dissimilarity      Method  s.e.   LCL   UCL Diversity Reftime
#> 1 esophagus_BC       0 0.500 13.452         0.122 Rarefaction 0.032 0.060 0.184        PD       1
#> 2 esophagus_BC       0 0.525 14.783         0.125 Rarefaction 0.033 0.061 0.189        PD       1
#> 3 esophagus_BC       0 0.550 16.277         0.128 Rarefaction 0.034 0.062 0.194        PD       1
#> 4 esophagus_BC       0 0.575 17.962         0.132 Rarefaction 0.035 0.064 0.200        PD       1
#> 5 esophagus_BC       0 0.600 19.904         0.136 Rarefaction 0.036 0.065 0.208        PD       1
#> 6 esophagus_BC       0 0.625 22.170         0.142 Rarefaction 0.038 0.068 0.216        PD       1
#> 
#> $`1-V`
#>         Dataset Order.q    SC   Size Dissimilarity      Method  s.e.   LCL   UCL Diversity Reftime
#> 1 esophagus_BC       0 0.500 13.452         0.065 Rarefaction 0.018 0.029 0.100        PD       1
#> 2 esophagus_BC       0 0.525 14.783         0.067 Rarefaction 0.019 0.030 0.103        PD       1
#> 3 esophagus_BC       0 0.550 16.277         0.068 Rarefaction 0.019 0.030 0.106        PD       1
#> 4 esophagus_BC       0 0.575 17.962         0.071 Rarefaction 0.020 0.031 0.110        PD       1
#> 5 esophagus_BC       0 0.600 19.904         0.073 Rarefaction 0.021 0.031 0.115        PD       1
#> 6 esophagus_BC       0 0.625 22.170         0.076 Rarefaction 0.022 0.032 0.120        PD       1
#> 
#> $`1-S`
#>         Dataset Order.q    SC   Size Dissimilarity      Method  s.e.   LCL   UCL Diversity Reftime
#> 1 esophagus_BC       0 0.500 13.452         0.122 Rarefaction 0.032 0.060 0.184        PD       1
#> 2 esophagus_BC       0 0.525 14.783         0.125 Rarefaction 0.033 0.061 0.189        PD       1
#> 3 esophagus_BC       0 0.550 16.277         0.128 Rarefaction 0.034 0.062 0.194        PD       1
#> 4 esophagus_BC       0 0.575 17.962         0.132 Rarefaction 0.035 0.064 0.200        PD       1
#> 5 esophagus_BC       0 0.600 19.904         0.136 Rarefaction 0.036 0.065 0.208        PD       1
#> 6 esophagus_BC       0 0.625 22.170         0.142 Rarefaction 0.038 0.068 0.216        PD       1
```

The output contains seven data frames: `gamma`, `alpha`, `beta`, `1-C`, `1-U`, `1-V`, `1-S`. For each data frame, it includes the diversity estimate (`Gamma`, `Alpha`, `Beta`, `Dissimilarity`), the diversity order (`Order.q`), `Method` (Interpolated, Observed, or Extrapolated, depending on whether the size `Size` is less than, equal to, or greater than the reference sample size), the sample coverage estimate (`SC`), the sample size (`Size`), the standard error from bootstrap replications (`s.e.`), the 95% lower and upper confidence limits of diversity (`LCL`, `UCL`), and the name of data set (`Dataset`). These diversity estimates with confidence intervals are used for plotting the diversity curve.

GRAPHIC DISPLAYS: FUNCTION ggiNEXTseq()

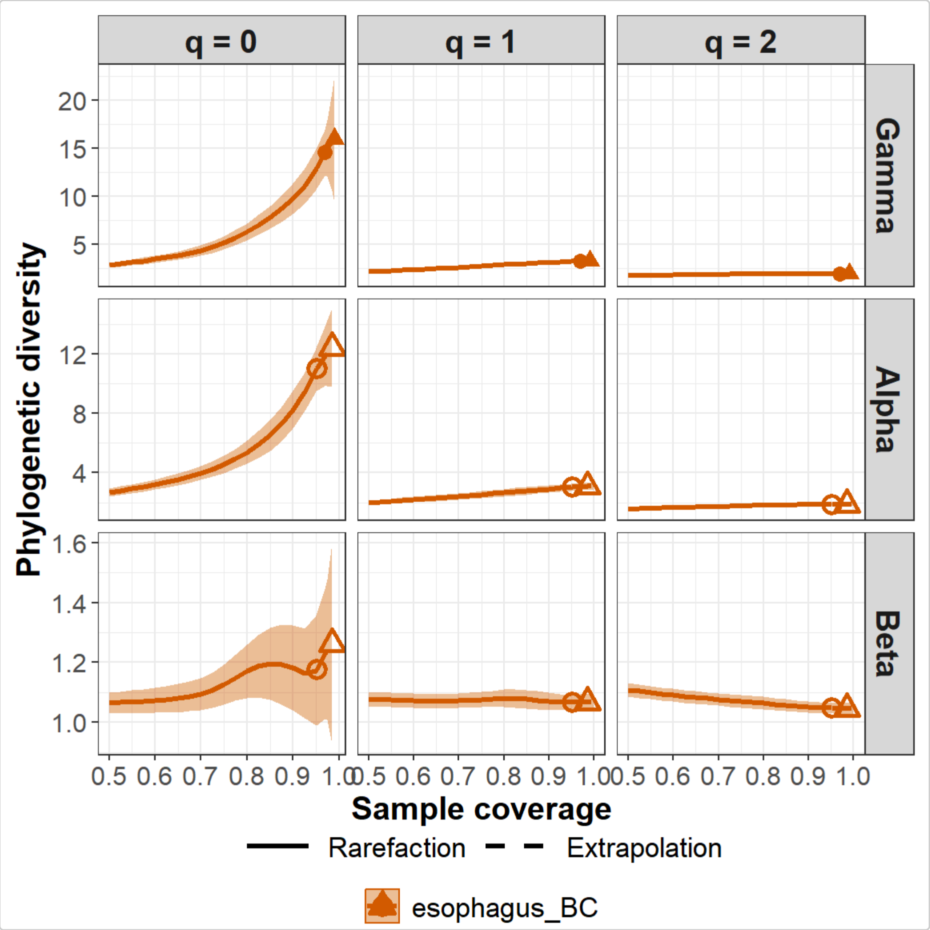
The function `ggiNEXTseq()`, which extends `ggplot2` to the `"iNEXTseq"` object with default arguments, is described as follows:

```
ggiNEXTseq(output, type = "B")
```

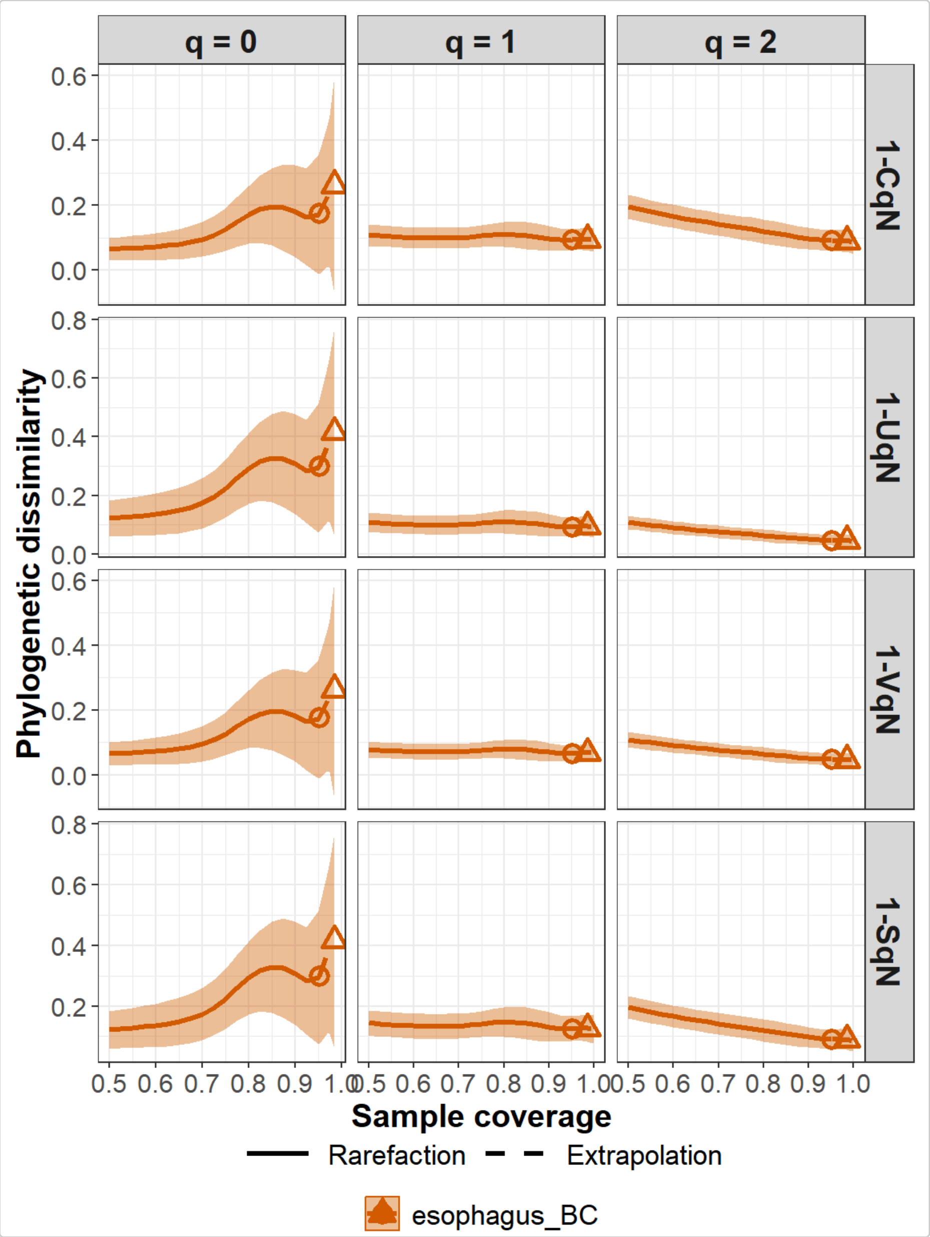
Argument	Description
`output`	the output of `iNEXTseq`.
`type`	(required only when base = “coverage”), selection of plot type: `type = “B”` for plotting the gamma, alpha, and beta diversity; `type = “D”` for plotting 4 turnover dissimilarities.

The `ggiNEXTseq()` function is a wrapper around the `ggplot2` package to create a R/E curve using a single line of code. The resulting object is of class `"ggplot"`, so it can be manipulated using the `ggplot2` tools. Users can visualize the output of beta diversity or four dissimilarities by setting the parameter **`type`**:

```
out = iNEXTseq(data = esophagus, q = c(0, 1, 2), nboot = 10,
               PDtree = esophagus_tree, PDreftime = NULL)
ggiNEXTseq(out, type = "B")
```

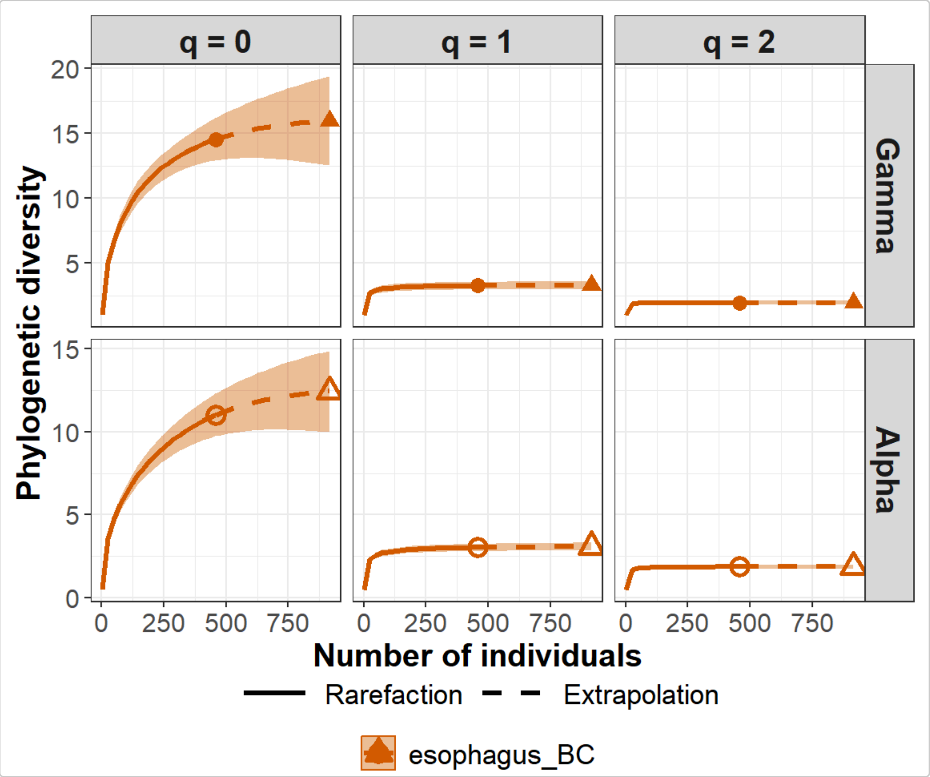


```
ggiNEXTseq(out, type = "D")
```



The following commands return the size-based R/E sampling curves for gamma and alpha diversity:

```
out_size = iNEXTseq(data = esophagus[1], q = c(0, 1, 2), base = "size",
                    nboot = 10, PDtree = esophagus_tree, PDreftime = NULL)
ggiNEXTseq(out_size)
```



MAIN FUNCTION: ObsAsyPD()

We first describe the main function `ObsAsyPD()` with default arguments:

```
ObsAsyPD(data, q = seq(0, 2, 0.2), weight = "size", nboot = 10, conf = 0.95,
         PDtree, type = "mle", decomposition = "relative")
```

The arguments of this function are briefly described below, and will be explained in more details by illustrative examples in later text. This main function computes observed and asymptotic diversity of order q between 0 and 2 (in increments of 0.2) for multiple-assemblage phylogenetic diversity; these values with different order q can be used to depict a q-profile in the `ggObsAsyPD` function.

Argument	Description
`data`	OTU count data can be input as a `matrix/data.frame` (species by assemblages), or a list of `matrices/data.frames`, each matrix represents species-by-assemblages abundance matrix.
`q`	a numerical vector specifying the diversity orders. Default is `seq(0, 2, 0.2)`.
`weight`	(required only when `type = "mle"` and `decomposition = "relative"`) weight for relative decomposition empirical estimate. Select size-weighted (`"size"`), equal-weighted (`"equal"`) or a numerical vector for weight. Default is `"size"`.
`nboot`	a positive integer specifying the number of bootstrap replications when assessing sampling uncertainty and constructing confidence intervals. Bootstrap replications are generally time consuming. Enter `0` to skip the bootstrap procedures. Default is `10`. Note that large bootstrap replication needs more run time.
`conf`	a positive number < 1 specifying the level of confidence interval. Default is `0.95`.
`PDtree`	a `phylo`, a phylogenetic tree in Newick format for all observed species in the pooled assemblage.
`type`	estimate type: empirical (`type = "mle"`) or asymptotic estimate (`type = "est"`). Default is `"mle"`.
`decomposition`	decomposition type: relative (`decomposition = "relative"`) or absolute decomposition (`decomposition = "absolute"`). Default is `"relative"`.

This function returns an `"ObsAsyPD"` object which can be further used to make plots using the function `ggObsAsyPD()` to be described below.

Examples

Run the `ObsAsyPD()` function with `esophagus` data to compute empirical estimate for relative decomposition of multiple-assemblage phylogenetic diversity. (Here we only show the first fourteen rows for output data frame)

```
data("esophagus")
data("esophagus_tree")

ObsAsyPD_out = ObsAsyPD(data = esophagus[1], q = seq(0, 2, 0.2), nboot = 10,
                        PDtree = esophagus_tree)
```

```
#> $esophagus_BC
#>         Dataset Method Order.q Estimator Bootstrap S.E.    LCL    UCL Decomposition
#> 1  esophagus_BC  Gamma     0.0    14.550          0.827 12.929 16.172      relative
#> 2  esophagus_BC  Alpha     0.0    11.174          0.488 10.218 12.130      relative
#> 3  esophagus_BC   Beta     0.0     1.302          0.048  1.207  1.397      relative
#> 4  esophagus_BC 1-CqN*     0.0     0.311          0.047  0.219  0.403      relative
#> 5  esophagus_BC 1-UqN*     0.0     0.471          0.049  0.374  0.568      relative
#> 6  esophagus_BC 1-VqN*     0.0     0.311          0.047  0.219  0.403      relative
#> 7  esophagus_BC 1-SqN*     0.0     0.471          0.049  0.374  0.568      relative
#> 8  esophagus_BC  Gamma     0.2    10.288          0.593  9.125 11.451      relative
#> 9  esophagus_BC  Alpha     0.2     8.256          0.416  7.441  9.071      relative
#> 10 esophagus_BC   Beta     0.2     1.246          0.040  1.168  1.325      relative
#> 11 esophagus_BC 1-CqN*     0.2     0.266          0.040  0.188  0.344      relative
#> 12 esophagus_BC 1-UqN*     0.2     0.385          0.045  0.296  0.473      relative
#> 13 esophagus_BC 1-VqN*     0.2     0.253          0.039  0.176  0.329      relative
#> 14 esophagus_BC 1-SqN*     0.2     0.400          0.046  0.311  0.489      relative
```

The output contains a data frames, it includes the estimate (`Estimator`) of diversity (`Gamma`, `Alpha` and `Beta`) and four types dissimilarity measure (`1-CqN`, `1-UqN`, `1-VqN`, `1-SqN`), the diversity order (`Order.q`), the standard error from bootstrap replications (`Bootstrap S.E.`), the 95% lower and upper confidence limits of diversity (`LCL`, `UCL`), beta diversity decomposition type (`Decomposition`), and the name of data set (`Dataset`). These diversity estimates with confidence intervals are used for plotting the diversity curve.

GRAPHIC DISPLAYS: FUNCTION ggObsAsyPD()

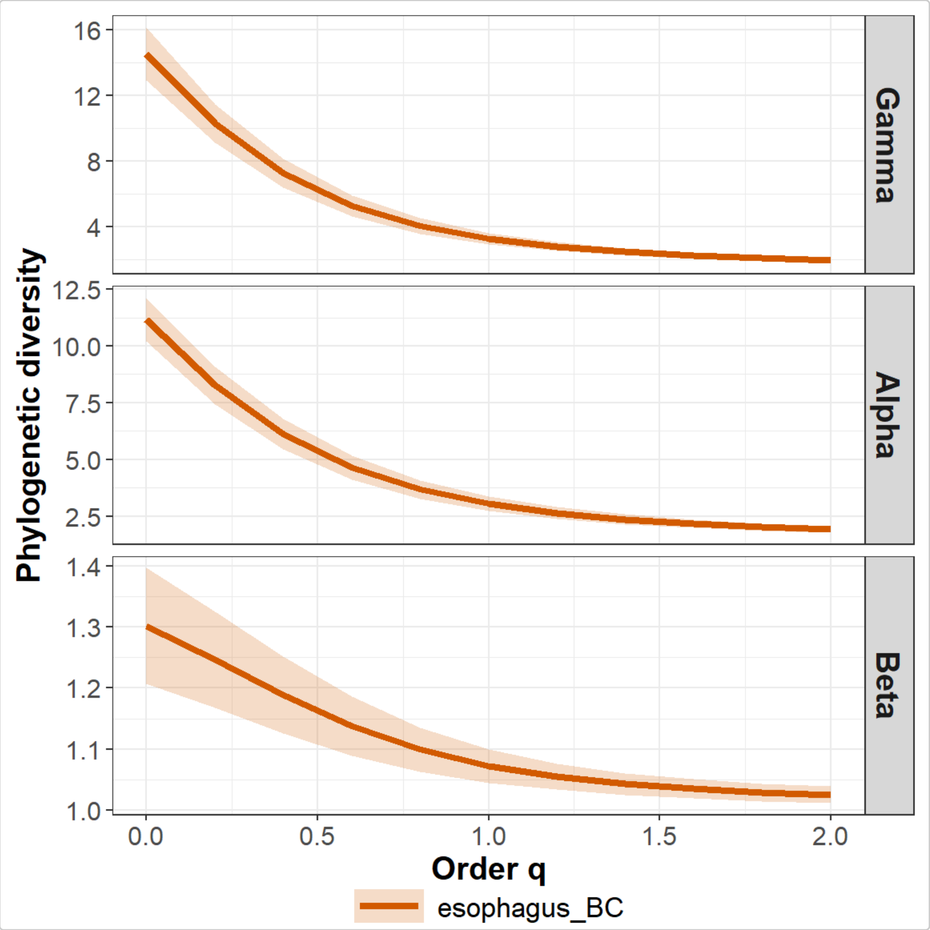
The function `ggObsAsyPD()`, which extends `ggplot2` to the `"ObsAsyPD"` object with default arguments, is described as follows:

```
ggObsAsyPD(output, type = "B")
```

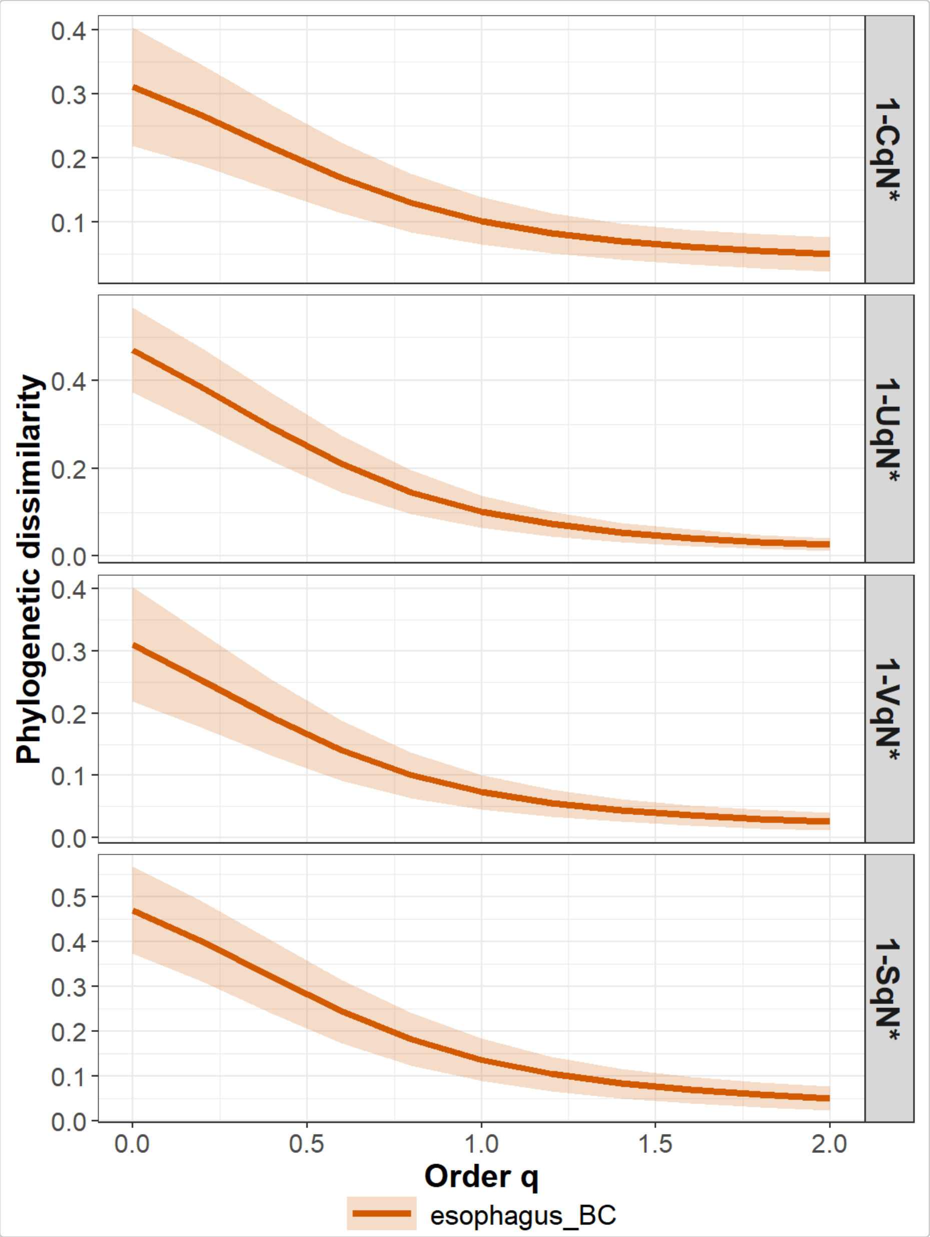
Argument	Description
`output`	the output of `ObsAsyPD`.
`type`	selection of plot type: `type = "B"` for plotting the gamma, alpha, and beta diversity; `type = "D"` for plotting 4 turnover dissimilarities.

The `ggObsAsyPD()` function is a wrapper around the `ggplot2` package to create a q-profile using a single line of code. The resulting object is of class `"ggplot"`, so it can be manipulated using the `ggplot2` tools. Users can visualize the output of beta diversity or four dissimilarities by setting the parameter **`type`**:

```
ObsAsyPD_out = ObsAsyPD(data = esophagus[1], q = seq(0, 2, 0.2), nboot = 10,
                        PDtree = esophagus_tree)
ggObsAsyPD(ObsAsyPD_out, type = "B")
```



```
ggObsAsyPD(ObsAsyPD_out, type = "D")
```



ANALYSIS FOR HIERARCHICAL STRUCTURE DATA

DATA INPUT FORMAT

Individual-based OTU count data

Input data for each data set with several assemblages/sites include samples species abundances in an empirical sample of n individuals ("reference sample"). When there are N assemblages in a data set, input data consist of a list with an S by N abundance matrix.

A data set (a small example dataset from Chinese wetlands community) is included in `iNEXT.seq` package for illustration. The data consist a data.frames of five assemblages/habitats ("NE", "NW", "NC", "YML" and "SC").

Run the following code to `wetland` OTU count data: (Here we only show the first ten rows for the matrix)

```
data("wetland")
wetland
```

```
#>             NE  NW   NC  YML   SC
#> OTU_32     132  10   17    0   32
#> OTU_50      12   4    9    1   37
#> OTU_305     15   0    1    0   18
#> OTU_408     14   4   24   10   23
#> OTU_6426    12   7   26    2  153
#> OTU_75       9   9    7    1   80
#> OTU_356   13  7   11   0  19
#> OTU_1    242 71 1584  11  27
#> OTU_41    22  3   13   2  34
#> OTU_59    23 51    4   0  32
```

Phylogenetic tree for phylogenetic diversity

To perform phylogenetic diversity analysis, the phylogenetic tree (in Newick format) spanned by species observed in the pooled data is required. For the data set `wetland`, the phylogenetic tree for all observed species (including species in “NE”, “NW”, “NC”, “YML” and “SC”) is stored in the file `wetland_tree`. A partial list of the tip labels and node labels (not required) are shown below.

```
data("wetland_tree")
wetland_tree
```

```
#> 
#> Phylogenetic tree with 404 tips and 403 internal nodes.
#> 
#> Tip labels:
#>   OTU_3829, OTU_705, OTU_3570, OTU_18, OTU_2733, OTU_108, ...
#> 
#> Rooted; includes branch lengths.
```

Structure matrix for hierarchical phylogenetic diversity

In addition to OTU count data and phylogenetic tree, a hierarchical structure matrix of data is required. The structure of matrix is m x N (number of hierarchical layers times number of assemblages). The hierarchical structure matrix of data set `wetland` are shown below.

```
data("wetland_mat")
wetland_mat
```

```
#>      [,1]  [,2]  [,3]  [,4]  [,5] 
#> [1,] "All" "All" "All" "All" "All"
#> [2,] "IW"  "IW"  "CW"  "CW"  "CW" 
#> [3,] "NE"  "NW"  "NC"  "YML" "SC"
```

MAIN FUNCTION: hierPD()

We first describe the main function `hierPD()` with default arguments:

```
hierPD(data, mat, q = seq(0, 2, 0.2), weight = "size", nboot = 10, conf = 0.95,
       PDtree, type = "mle", decomposition = "relative")
```

The arguments of this function are briefly described below, and will be explained in more details by illustrative examples in later text. This main function computes observed and asymptotic diversity of order q between 0 and 2 (in increments of 0.2) for hierarchical phylogenetic diversity; these values with different order q can be used to depict a q-profile in the `gghierPD` function.

Argument	Description
`data`	data should be input as a `matrix/data.frame` (species by assemblages).
`mat`	hierarchical structure of data should be input as a `matrix`.
`q`	a numerical vector specifying the diversity orders. Default is `seq(0, 2, 0.2)`.
`weight`	(required only when `type = “mle”` and `decomposition = “relative”`) weight for relative decomposition empirical estimate. Select size-weighted (`“size”`), equal-weighted (`“equal”`) or a numerical vector for weight. Default is `“size”`.
`nboot`	a positive integer specifying the number of bootstrap replications when assessing sampling uncertainty and constructing confidence intervals. Bootstrap replications are generally time consuming. Enter `0` to skip the bootstrap procedures. Default is `10`. Note that large bootstrap replication needs more run time.

Argument	Description
conf	a positive number < 1 specifying the level of confidence interval. Default is 0.95.
PDtree	a phylo, a phylogenetic tree in Newick format for all observed species in the pooled assemblage.
type	estimate type: empirical (type = “mle”) or asymptotic estimate (type = “est”). Default is “mle”.
decomposition	decomposition type: relative (decomposition = “relative”) or absolute decomposition (decomposition = “absolute”). Default is “relative”.

This function returns an "hierPD" object which can be further used to make plots using the function gghierPD() to be described below.

Examples

Run the hierPD() function with wetland data to compute empirical estimate for relative decomposition of hierarchical phylogenetic diversity. (Here we only show the first fifteen rows for output data frame)

```
data("wetland")
data("wetland_mat")
data("wetland_tree")

hierPD_out = hierPD(data = wetland, mat = wetland_mat, q = seq(0, 2, 0.2),
                    nboot = 10, PDtree = wetland_tree)
```

```
#>            Method Order.q Estimator Bootstrap S.E.    LCL    UCL Decomposition
#> 10     qPD_alpha1       0    82.280          0.550 81.202 83.358      relative
#> 11     qPD_alpha2       0    87.694          0.527 86.662 88.726      relative
#> 12      qPD_gamma       0    89.618          0.000 89.618 89.618      relative
#> 13     qPD_Beta 1       0     1.066          0.002  1.061  1.070      relative
#> 14     qPD_Beta 2       0     1.022          0.006  1.010  1.034      relative
#> 15 qPD_Beta_max 1       0     2.362          0.010  2.343  2.382      relative
#> 16 qPD_Beta_max 2       0     2.000          0.000  2.000  2.000      relative
#> 17    1-CqN*(1|2)       0     0.048          0.002  0.045  0.052      relative
#> 18    1-CqN*(2|3)       0     0.022          0.006  0.010  0.034      relative
#> 19    1-UqN*(1|2)       0     0.107          0.003  0.100  0.114      relative
#> 20    1-UqN*(2|3)       0     0.043          0.012  0.020  0.066      relative
#> 21    1-VqN*(1|2)       0     0.048          0.002  0.045  0.052      relative
#> 22    1-VqN*(2|3)       0     0.022          0.006  0.010  0.034      relative
#> 23    1-SqN*(1|2)       0     0.107          0.003  0.100  0.114      relative
#> 24    1-SqN*(2|3)       0     0.043          0.012  0.020  0.066      relative
```

The output contains a data frames, it includes the estimate (Estimator) of hierarchical diversity (qPD_gamma, qPD_alpha, qPD_Beta and qPD_Beta_max) and four types dissimilarity measure (1-CqN, 1-UqN, 1-VqN, 1-SqN), the diversity order (Order.q), the standard error from bootstrap replications (Bootstrap S.E.), the 95% lower and upper confidence limits of diversity (LCL, UCL), and beta diversity decomposition type (Decomposition). These diversity estimates with confidence intervals are used for plotting the diversity curve.

GRAPHIC DISPLAYS: FUNCTION gghierPD()

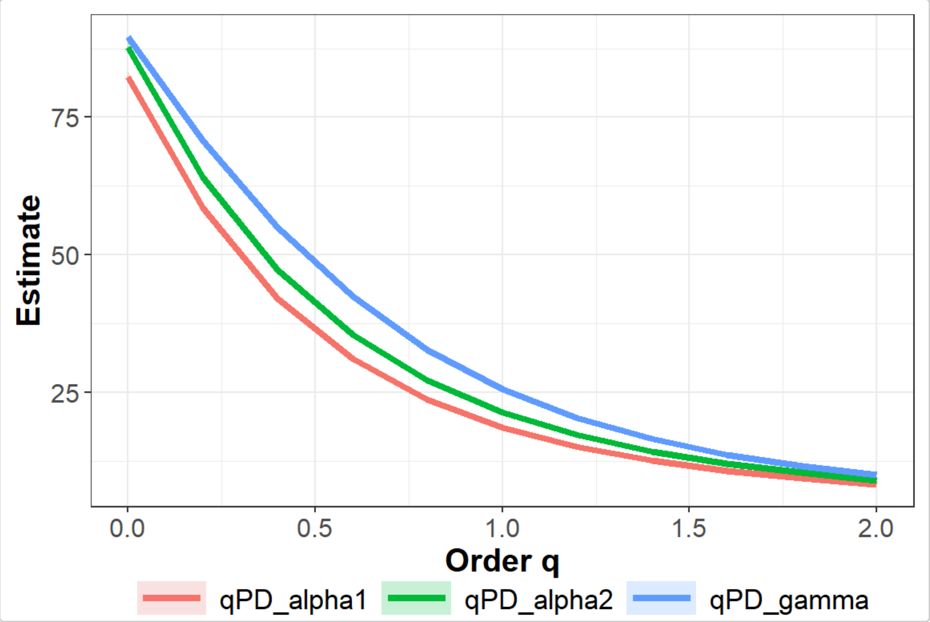
The function gghierPD(), which extends ggplot2 to the "hierPD" object with default arguments, is described as follows:

```
gghierPD(output, type = "A")
```

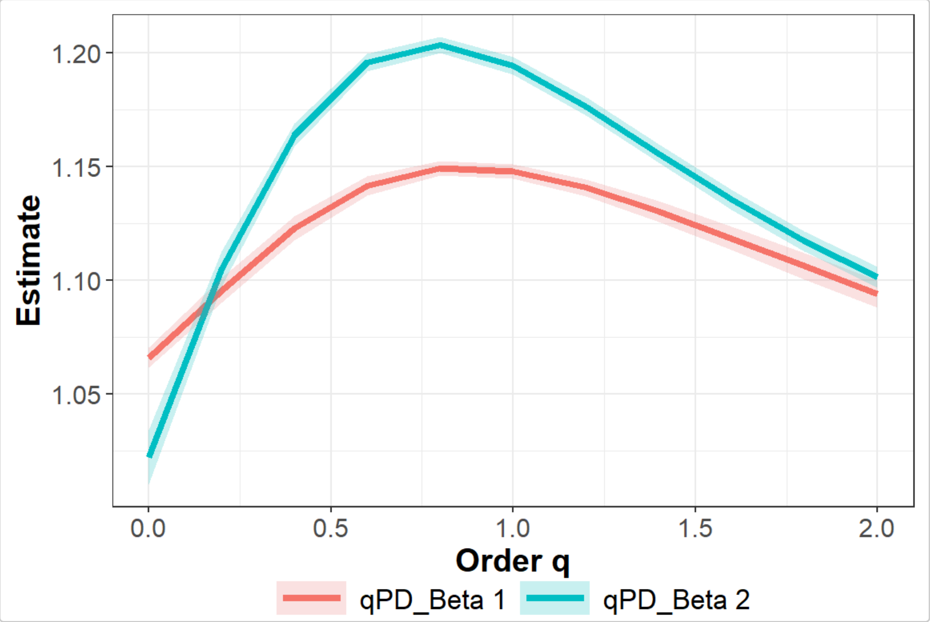
Argument	Description
output	the output of hierPD.
type	selection of plot type: (type = “A”) for alpha and gamma diversity; (type = “B”) for beta diversity; (type = “D”) for dissimilarity measure based on multiplicative decomposition.

The gghierPD() function is a wrapper around the ggplot2 package to create a q-profile using a single line of code. The resulting object is of class "ggplot", so it can be manipulated using the ggplot2 tools. Users can visualize the output of beta diversity or four dissimilarities by setting the parameter `type`:

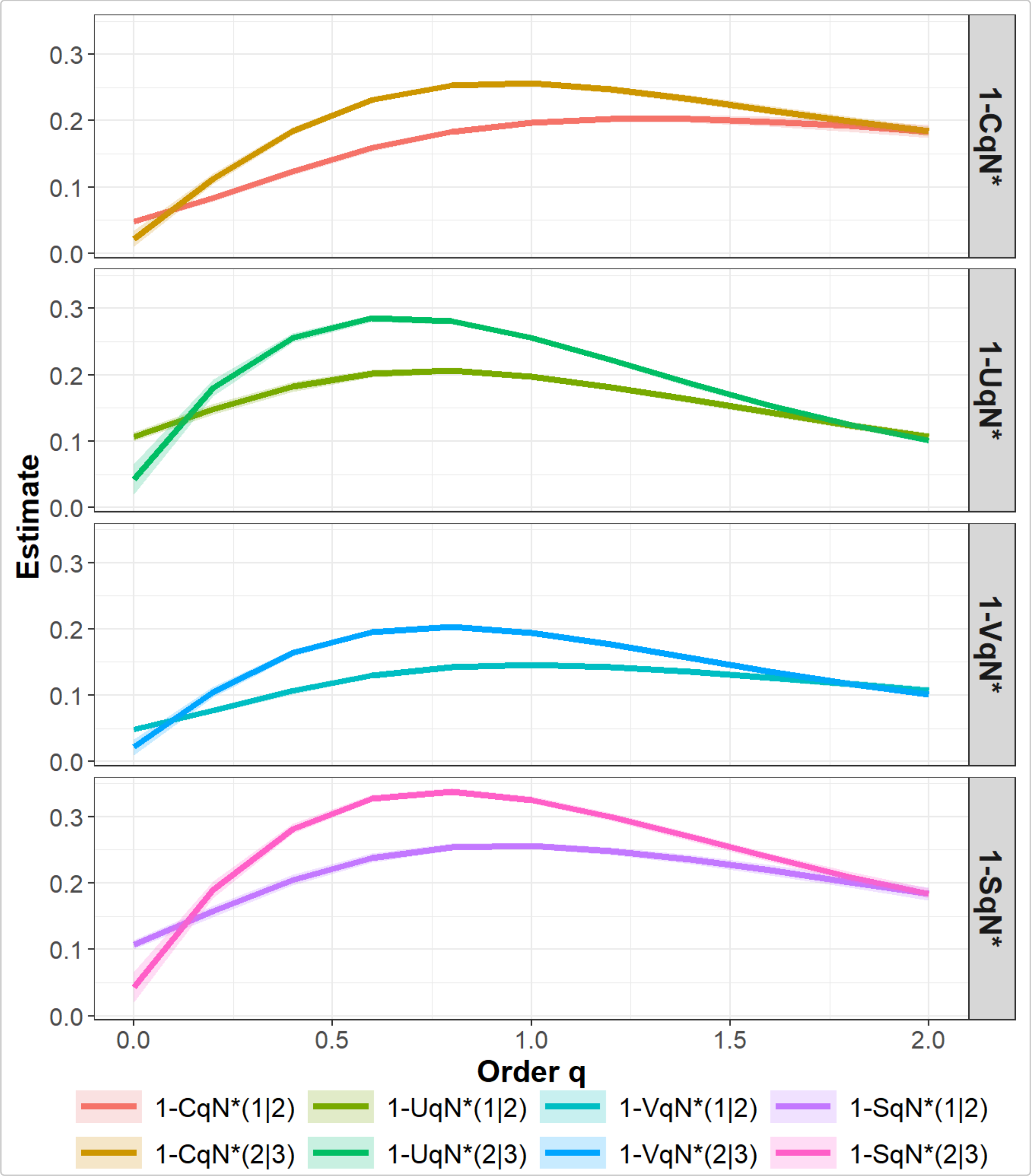
```
hierPD_out = hierPD(data = wetland, mat = wetland_mat, q = seq(0, 2, 0.2),
                    nboot = 10, PDtree = wetland_tree)
gghierPD(hierPD_out, type = "A")
```



```
gghierPD(hierPD_out, type = "B")
```



```
gghierPD(hierPD_out, type = "D")
```



References

- Chiu, C.-H., Jost, L. and Chao*, A. (2014). Phylogenetic beta diversity, similarity, and differentiation measures based on Hill numbers. *Ecological Monographs*, 84, 21-44.
- Routledge, R. (1979). Diversity indices: which ones are admissible? *Journal of Theoretical Biology*, 76(4), 503-515.