Package: MF.beta4 (via r-universe)

September 14, 2024

Type Package

Title Measuring Ecosystem Multi-Functionality and Its Decomposition

Version 1.1.0

Author Anne Chao [aut, cre], Chun-Yu Liu [ctb], KaiHsiang Hu [ctb]

Maintainer Anne Chao <chao@stat.nthu.edu.tw>

URL https://github.com/AnneChao/MF.beta4

BugReports https://github.com/AnneChao/MF.beta4/issues

Description Provide simple functions to (i) compute a class of multi-functionality measures for a single ecosystem for given function weights, (ii) decompose gamma multi-functionality for multiple ecosystems into a within-ecosystem component (alpha multi-functionality) and an among-ecosystem component (beta multi-functionality). In each case, the correlation between functions can be corrected for. Based on biodiversity and ecosystem function data, this software also facilitates graphics for assessing biodiversity-ecosystem functioning relationships across scales.

License GPL (>= 3)

Depends R (>= 4.0)

Imports broom, devtools, ggplot2, dplyr, stats, ggpubr, grid, purrr, patchwork, tidyr, lme4, lmerTest, tidyverse, reshape2

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

ByteCompile true

Repository https://annechao.r-universe.dev

RemoteUrl https://github.com/annechao/mf.beta4

RemoteRef HEAD

RemoteSha 4f5727032977dcfd099424786b7d1cff13f66bc6

Contents

`MF.beta4-package` *Measuring ecosystem multifunctionality and assessing BEF relationships*

Description

`MF.beta4` is an R package for measuring ecosystem multifunctionality and assessing BEF relationships. The measures are illustrated by using ecosystem function and biodiversity data collected in a total of 209 plots in six European countries (the FunDivEUROPE dataset). All data are available from the Dryad repository; see Ratcliffe et al. (2017a, b) and Scherer-Lorenzen et al. (2023) for data details. The software was originally developed for the Beta4 project (Müller et al. 2022) on the effect of enhancing the beta diversity between forest patches on ecosystem multifunctionality and forest resilience across spatial scales.

Based on a framework of Hill-Chao numbers of orders q = 0, 1 and 2, `MF.beta4` features the following multifunctionality measures for a single and multiple ecosystems; see Chao et al. (2023) for pertinent methodology and decomposition theory.

(1) Multifunctionality measures in a single ecosystem: `MF.beta4` computes a class of weighted multifunctionality measures for given function weights. Multifunctionality measures that correct for strong correlations between ecosystem functions to avoid redundancy are also provided.

(2) Multifunctionality measures in multiple ecosystems: for given function weights, `MF.beta4` computes the gamma multifunctionality of pooled ecosystems, the within-ecosystem component (alpha multifunctionality) and the among-ecosystem component (beta multifunctionality). The correlation between functions can also be corrected for.

Based on biodiversity and function data from ecosystems, this package also provides graphics for assessing biodiversity-ecosystem functioning (BEF) relationships across scales.

This package includes four main functions:

1. `function_normalization` transforms ecosystem functions data to values between 0 and 1.

2. `MF1_single` computes a class of weighted multifuctionality measures in a single ecosystem for given individual function weights separately for two cases: (i) correlations between functions are not corrected for, and (ii) correlations between functions are corrected for.

3. `MF2_multiple` computes alpha, beta and gamma multifuctionality measures in multiple ecosystems for given function weights separately for two cases (i) correlations between functions are not corrected for, and (ii) correlations between functions are corrected for.

4. `MFggplot` provides the graphical BEF relationships based on the output from the function `MF1_single` or `MF2_multiple`.

Author(s)

Anne Chao, Chun-Yu Liu, K. H. Hu
Maintainer: Anne Chao <chao@stat.nthu.edu.tw>

References

Chao, A., Chiu, C. H., Hu, K. H., van der Plas, F., Cadotte, M. W., Mitesser, O., et al. (2023). Hill-Chao numbers in multifunctionality allows decomposing gamma multifunctionality into alpha and beta components. To appear in Ecology Letters.

Müller, J., Mitesser, O. Cadotte, M. W., van der Plas, F., Mori, A, Ammer, C., Eisenhauer N. (2023). Enhancing the structural diversity between forest patches - a concept and real-world experiment to study biodiversity and multifunctionality across spatial scales. Global Change Biology, 29, 1437-1450.

Ratcliffe, S. Wirth, C., Jucker, T. van der Plas, F., Scherer-Lorenzen, M. Verheyen, K. et al. (2017a). Biodiversity and ecosystem functioning relations in European forests depend on environmental context. Ecology Letters, 20, 1414-1426.

Ratcliffe, S. Wirth, C., Jucker, T., van der Plas, F., Scherer-Lorenzen, M., Verheyen, K. et al. (2017b). Data for Biodiversity and ecosystem functioning relations in European forests depend on environmental context. https://doi.org/10.6084/m9.figshare.5368846.v1

Scherer-Lorenzen, M. et al. (2023). The functional significance of tree species diversity in European forests - the FunDivEUROPE dataset [Dataset]. Dryad. https://doi.org/10.5061/dryad.9ghx3ffpz

---

`forest_biodiversity_data`

Biodiversity data for six European forests

Description

In addition to row and column names, this dataset consists of three columns: the "plotID" column indicates the names of plots, the "species" column includes species names, and the column "abundance" (basal area as a proxy of species abundance). Because missing values of "basal area" in the original dataset were imputed by the mean of the same species within the country, and basal areas were combined for two species ("Betula pendula" and "Betula pubescens"), the dataset provided with the package is slightly different from the original dataset provided in Scherer-Lorenzen et al. (2023).

Usage

```
data("forest_biodiversity_data")
```

Format

a data.frame with 481 rows (the total number of combinations of plot and tree species in 209 plots) and 3 columns (plotID, species name, and the corresponding basal area as species abundance).

References

Scherer-Lorenzen, M. et al. (2023). The functional significance of tree species diversity in European forests - the FunDivEUROPE dataset [Dataset]. Dryad. https://doi.org/10.5061/dryad.9ghx3ffpz

---

forest_function_data_normalized

Normalized ecosystem function data for six European forests

Description

This dataset represents the normalized version of the raw dataset (`forest_function_data_raw`). All raw ecosystem functions are normalized to the range of [0, 1], whereas other variables remain unchanged.

Usage

```
data("forest_function_data_normalized")
```

Format

a data.frame with 209 plots (rows) and 32 columns, in addition to row and column names; the first 5 columns show the relevant plot information, followed by 26 normalized ecosystem functions. The last column shows the corresponding country for each plot.

`forest_function_data_raw` *Ecosystem function data for six European forests*

Description

In addition to plot information, this dataset includes raw values of 26 ecosystem functions collected from 209 plots (each with 30 m × 30 m) in six European countries, representing six major European forest types: boreal forest (Finland, 28 plots); hemi-boreal (Poland, 43 plots); temperate deciduous (Germany, 38 plots); mountainous deciduous (Romania, 28 plots); thermophilous deciduous (Italy, 36 plots); and Mediterranean mixed (Spain, 36 plots). See Table 1 of Ratcliffe et al. (2017a) for a description of the 26 functions. Each plot is designated as an ecosystem in assessing BEF relationships. See Ratcliffe et al. (2017b) and Scherer-Lorenzen et al. (2023) for the original dataset. For each missing value of functions in the original dataset, the mean of the given function within the country was imputed. An additional column "country" for each plot is added (as the last column) because function normalization and relevant analyses will be performed within each country. Thus, the dataset provided with the package is slightly different from the original one.

Usage

```
data("forest_function_data_raw")
```

Format

a data.frame with 209 plots (rows) and 32 columns, in addition to row and column names; the first 5 columns show the relevant plot information, followed by 26 raw ecosystem functions (columns 6 to 31). The last column shows the corresponding country for each plot.

References

Ratcliffe, S. Wirth, C., Jucker, T. van der Plas, F., Scherer-Lorenzen, M. Verheyen, K. et al. (2017a). Biodiversity and ecosystem functioning relations in European forests depend on environmental context. Ecology Letters, 20, 1414–1426.

Ratcliffe, S. Wirth, C., Jucker, T., van der Plas, F., Scherer-Lorenzen, M., Verheyen, K. et al. (2017b). Data for Biodiversity and ecosystem functioning relations in European forests depend on environmental con- text. https://doi.org/10.6084/m9.figshare.5368846.v1

Scherer-Lorenzen, M. et al. (2023). The functional significance of tree species diversity in European forests - the FunDivEUROPE dataset [Dataset]. Dryad. https://doi.org/10.5061/dryad.9ghx3ffpz

function_normalization *Normalize raw ecosystem function values to [0,1]*

Description

function_normalization transforms raw function values to values between 0 and 1. For positive functionality, ecosystems with the highest value in the raw function data are transformed to the maximal value of 1, and those with the lowest raw value are transformed to the minimum value of 0. Because the value "0" always implies absent functions, if the lowest raw value is not 0, the transformed 0 from this non-zero raw value will be replaced by a very small number,e.g., 10^(-5). In a similar manner, for negative functionality, if the highest raw value is not 0, the transformed 0 will also be replaced by a very small number, e.g., 10^(-5). These replacements will not affect any numerical computations but will help indicate that the transformed values represent functions that should be regarded as "present" ones. Thus, present or absent functions can be clearly distinguished in the transformed data, and the information on presence/absence of functions is required in the decomposition of multifunctionality among ecosystems.

Usage

```
function_normalization(
  data,
  fun_cols = 1:ncol(data),
  negative = NULL,
  by_group = NULL
)
```

Arguments

data	data can be input as a data.frame with ecosystems/plots as rows and relevant ecosystem/plot information and ecosystem functions as columns. All missing values should be imputed in the input data. If the stratifying/grouping variable (specified in the argument by_group) is not NULL, data must contain a column that is used for stratification.
fun_cols	the columns that represent ecosystem functions.
negative	names of the negative functionality.
by_group	the column name of the stratifying variable that is used to group data for performing normalization. For example, if by_group = "country", then all functions will be normalized to the range of [0, 1] within a country. Default is NULL.

Value

a data.frame with all values in functions (specified in fun_cols) being replaced by the transformed values between 0 and 1.

Examples

```
library(dplyr)

### Use data from six countries

data("forest_function_data_raw")
function_normalization(data = forest_function_data_raw, fun_cols = 6:31,
                       negative = c("soil_cn_ff_10","wue"), by_group = "country")


### Use partial data to quickly obtain output
### (Take the first 18 plots in Germany and the last 18 plots in Italy)

data("forest_function_data_raw")
GER_ITA_forest_function_raw <- filter(forest_function_data_raw,
                                      country=="GER"|country=="ITA")[c(1:18,57:74),]
function_normalization(data = GER_ITA_forest_function_raw, fun_cols = 6:31,
                       negative = c("soil_cn_ff_10","wue"), by_group = "country")
```

`MF1_single` *multifunctionality measures for a single ecosystem*

Description

`MF1_single` computes multifunctionality measures of orders q = 0, 1 and 2 for given function weights in a single ecosystem separately for two cases (i) correlations between functions are not corrected for, and (ii) correlations between functions are corrected for.

Usage

```
MF1_single(func_data, species_data = NULL, weight = 1, q = c(0, 1, 2))
```

Arguments

`func_data`	ecosystem function data should be input as a data.frame (ecosystems by functions). All function values must be normalized between 0 and 1. The row names of `func_data` should be set the same as the names of plotID specified in `species_data` if `species_data` is not `NULL`.
`species_data`	species abundance data should be input as a data.frame and must include three columns: "plotID", "species" and "abundance" (or any proxy such as basal area). Default is `NULL`.
`weight`	a constant number (if all weights are equal) or a numerical vector specifying weights for ecosystem functions. In the latter case, the length of `weight` must be equal to the number of functions. Default is `weight = 1`, which means equal weight and weight = 1 for all ecosystem functions.
`q`	a numerical vector specifying the multifunctionality and diversity orders. Default is q = 0, 1 and 2.

Value

a data.frame with columns "plotID", "Type" (corr_uncorrected or corr_corrected), "Order.q" and "qMF" (multifunctionality of order q). When species_data is not NULL, the data.frame will include an additional column "Species.diversity" in the last column.

Examples

```
library(dplyr)


### Use data from six countries

data("forest_function_data_normalized")
data("forest_biodiversity_data")
MF1_single(func_data = forest_function_data_normalized[,6:31], weight = 1,
           species_data = forest_biodiversity_data)


### Use partial data to quickly obtain output
### (Take the first 18 plots in Germany and the last 18 plots in Italy)

data("forest_function_data_raw")
data("forest_biodiversity_data")
GER_ITA_forest_function_raw <- filter(forest_function_data_raw,
                                      country=="GER"|country=="ITA")[c(1:18,57:74),]
GER_ITA_forest_function_normalized <- function_normalization(data = GER_ITA_forest_function_raw,
                                                             fun_cols = 6:31,
                                                    negative = c("soil_cn_ff_10","wue"),
                                                             by_group = "country")
GER_ITA_forest_biodiversity <- forest_biodiversity_data[c(49:82,181:229),]
MF1_single(func_data = GER_ITA_forest_function_normalized[,6:31], weight = 1,
           species_data = GER_ITA_forest_biodiversity)
```

MF2_multiple *multifunctionality measures for multiple ecosystems*

Description

MF2_multiple computes alpha, beta and gamma multifuctionality measures of orders q = 0, 1 and 2 for given function weights in multiple ecosystems separately for two cases (i) correlations between functions are not corrected for, and (ii) correlations between functions are corrected for.

Usage

```
MF2_multiple(
  func_data,
  species_data = NULL,
  weight = 1,
```

```
  q = c(0, 1, 2),
  by_group = NULL,
  by_pair = TRUE
)
```

Arguments

func_data	ecosystem function data should be input as a data.frame (ecosystems by functions for multiple ecosystems). All function values must be normalized between 0 and 1. For by_group = NULL, the func_data must contain only the ecosystem function columns. (e.g., those columns specified in the argument fun_cols if user use function_normalization to do normalization). If by_group is not NULL,in addition to ecosystem function columns, the by_group column must be included. The row names of func_data should be set the same as the names of plotID specified in species_data if species_data is not NULL.
species_data	species abundance data should be input as a data.frame and must include three columns: "plotID", "species" and "abundance". Default is NULL. If by_group is not NULL and species_data is not NULL,in addition to the three columns mentioned earlier, the by_group column must be included.
weight	a constant number (if all weights are equal) or a numerical vector specifying weights for ecosystem functions. In the latter case, the length of weight must be equal to the number of functions. Default is weight = 1, which means equal weight and weight = 1 for all ecosystem functions.
q	a numerical vector specifying the multifunctionality and diversity orders. Default is q = 0, 1 and 2.
by_group	the column name of the stratifying variable that is used to group data for performing decomposition. For example, if by_group = "country", then multifunctionality decomposition is performed for any pair of plots selected within a country. The by_group setting must be the same as that set in function_normalization. Default is NULL.
by_pair	whether to calculate multifunctionality by pairs or not. Default is TRUE.

Value

a data.frame with columns "plotID" (combinations of plot pairs, if calculating not by pairs, then there is no such column), "Order.q" , "Type" (corr_uncorrected or corr_corrected) , "Scale" (gamma, alpha or beta) and "qMF" (multifunctionality of order q). When by_group is not NULL (i.e., the column name of the stratifying variable is specified), an additional column with stratification variable (e.g., "country" of the plot pairs) is also shown after the plotID column. For species_data is not NULL, the data.frame will show an additional column contain "Species.diversity" in the last column.

Examples

```
library(dplyr)
```

```
### Use data from five countries (data in Finland are excluded)

data("forest_function_data_normalized")
data("forest_biodiversity_data")
forest_function_data_normalized <- filter(forest_function_data_normalized, country != "FIN")
forest_biodiversity_data <- forest_biodiversity_data[-(1:48),]
forest_biodiversity_data <- forest_biodiversity_data %>% mutate(country = substr(plotID, 1, 3))

MF2_multiple(func_data = forest_function_data_normalized[,6:32],
             species_data = forest_biodiversity_data,
             weight = 1,
             by_group = "country")




### Use partial data to quickly obtain output
### (Take the first 18 plots in Germany and the last 18 plots in Italy)

data("forest_function_data_raw")
data("forest_biodiversity_data")
GER_ITA_forest_function_raw <- filter(forest_function_data_raw,
                                      country=="GER"|country=="ITA")[c(1:18,57:74),]
GER_ITA_forest_function_normalized <- function_normalization(data = GER_ITA_forest_function_raw,
                                                             fun_cols = 6:31,
                                                    negative = c("soil_cn_ff_10","wue"),
                                                             by_group = "country")
GER_ITA_forest_biodiversity <- forest_biodiversity_data[c(49:82,181:229),]
GER_ITA_forest_biodiversity <- GER_ITA_forest_biodiversity %>%
                                                   mutate(country = substr(plotID, 1, 3))

MF2_multiple(func_data = GER_ITA_forest_function_normalized[,6:32],
             species_data = GER_ITA_forest_biodiversity,
             weight = 1,
             by_group = "country")


### Use partial data to calculate multifunctionality by all plots in country, not by pairs
### (Take the first 3 plots in each country)

data("forest_function_data_raw")
data("forest_biodiversity_data")

forest_function_data_raw_3plots <- forest_function_data_raw[c(1:3,29:31,67:69,103:105,
                                                              146:148,174:176),]
forest_function_data_normalized_3plots <-
                    function_normalization(data = forest_function_data_raw_3plots,
                                           fun_cols = 6:31,
                                           negative = c("soil_cn_ff_10","wue"),
                                           by_group = "country")
forest_biodiversity_data_3plots <-
                forest_biodiversity_data[c(1:6,49:52,141:148,230:232,351:355,411:417),]
```

```
MF2_multiple(func_data = forest_function_data_normalized_3plots[,6:32],
             species_data = forest_biodiversity_data_3plots,
             weight = 1,
             by_group = "country", by_pair = FALSE)
```

MFggplot *ggplot2 extension for a MF1_single or MF2_multiple object*

Description

MFggplot provides graphical BEF relationships based on the output from the function MF1_single or MF2_multiple.

Usage

```
MFggplot(output, model = "LMM.both", caption = "slope", by_group = NULL)
```

Arguments

output	the output obtained from MF1_single or MF2_multiple. For output obtained from MF1_single, if BEF relationships are desired within each category specified by_group, the by_group column must be included in the input.
model	specifying the fitting model, model = "lm" for linear model; model = "LMM.intercept", "LMM.slope" and "LMM.both" for linear mixed models with random effects for intercepts, slopes, and both, respectively. Default is model = "LMM.both". If output is obtained from MF2_multiple with by_pair = "FALSE", model can only set as "lm".
caption	caption that will be shown in the BEF plots; caption = "slope" to show the estimated slopes in each plot, or caption = "R.squared" to show the ordinary R-squared for linear models or estimated marginal and conditional R-squared for linear mixed models in each plot. Default is caption = "slope".
by_group	the column name of the stratifying variable that is used to group data for model fitting. For example, if by_group = "country", then model will be fitted within each country. Default is NULL. It is required if a linear mixed model is selected in the model. If output is obtained from MF2_multiple with by_pair = "TRUE", the by_group setting must be the same as that set in MF2_multiple. If output is obtained from MF2_multiple with by_pair = "FALSE", the by_group should be NULL. Because the observations is not large enough to fit model in each group.

Value

For an MF1_single object, this function returns a figure that plots the BEF relationship between multifunctionality of order q (= 0, 1 and 2) and species diversity of the same order q for two cases (i) correlations between functions are not corrected for, and (ii) correlations between functions are corrected. The fitted lines for the chosen model are also shown in the figure.

For an MF2_multiple object of given individual function weights, this function returns a list of two objects ($corr_uncorrected and $corr_corrected) that respectively for two cases: (i) correlations between functions are not corrected for, and (ii) correlations between functions are corrected for.

Each object consists of four figures: "$ALL" returns a figure that depicts the BEF relationship between alpha/beta/gamma multifunctionality of order q (= 0, 1 and 2) and the corresponding species diversity of the same order q. The fitted lines for the chosen model are also shown in the figure. "$Gamma" returns only the gamma part of "$ALL", "$Alpha" returns only the alpha part of "$ALL", and "$Beta" returns only the beta part of "$ALL".

Examples

```
library(dplyr)

### Use data from six countries


## single ecosystem
data("forest_function_data_normalized")
data("forest_biodiversity_data")
output1 <- MF1_single(func_data = forest_function_data_normalized[,6:31], weight = 1,
                      species_data = forest_biodiversity_data)

## Display fitted line of linear mixed model with random slopes and random intercepts
output1 <- data.frame(output1, country=rep(forest_function_data_normalized$country, each = 6))
MFggplot(output1, model = "LMM.both", by_group="country", caption = "slope")


### Use data from five countries (data in Finland are excluded)

## multiple ecosystems
data("forest_function_data_normalized")
data("forest_biodiversity_data")
forest_function_data_normalized <- filter(forest_function_data_normalized, country != "FIN")
forest_biodiversity_data <- forest_biodiversity_data[-(1:48),]
output2 <- MF2_multiple(func_data = forest_function_data_normalized[,6:32],
                        species_data = forest_biodiversity_data,
                        weight = 1,
                        by_group = "country")

## Display fitted line of linear mixed model with random slopes and random intercepts
figure_LMM <- MFggplot(output2, model = "LMM.both", by_group = "country",
                       caption = "slope")
```

```
figure_LMM$corr_uncorrected$ALL
figure_LMM$corr_corrected$ALL


### Use partial data to quickly obtain output
### (Take the first 18 plots in Germany and the last 18 plots in Italy)

## single ecosystem
data("forest_function_data_raw")
data("forest_biodiversity_data")
GER_ITA_forest_function_raw <- filter(forest_function_data_raw,
                                      country=="GER"|country=="ITA")[c(1:18,57:74),]
GER_ITA_forest_function_normalized <- function_normalization(data = GER_ITA_forest_function_raw,
                                                             fun_cols = 6:31,
                                                    negative = c("soil_cn_ff_10","wue"),
                                                             by_group = "country")
GER_ITA_forest_biodiversity <- forest_biodiversity_data[c(49:82,181:229),]
output3 <- MF1_single(func_data = GER_ITA_forest_function_normalized[,6:31], weight = 1,
                      species_data = GER_ITA_forest_biodiversity)


## Display fitted line of linear mixed model with random slopes and random intercepts
output3 <- data.frame(output3, country=rep(GER_ITA_forest_function_normalized$country, each = 6))
MFggplot(output3, model = "LMM.both", by_group="country", caption = "slope")


## multiple ecosystems
data("forest_function_data_raw")
data("forest_biodiversity_data")
GER_ITA_forest_function_raw <- filter(forest_function_data_raw,
                                      country=="GER"|country=="ITA")[c(1:18,57:74),]
GER_ITA_forest_function_normalized <- function_normalization(data = GER_ITA_forest_function_raw,
                                                             fun_cols = 6:31,
                                                    negative = c("soil_cn_ff_10","wue"),
                                                             by_group = "country")
GER_ITA_forest_biodiversity <- forest_biodiversity_data[c(49:82,181:229),]
output4 <- MF2_multiple(func_data = GER_ITA_forest_function_normalized[,6:32],
                        species_data = GER_ITA_forest_biodiversity,
                        weight = 1,
                        by_group = "country")


## Display fitted line of linear mixed model with random slopes and random intercepts
figure_LMM_GER_ITA <- MFggplot(output4, model = "LMM.both", by_group = "country",
                               caption = "slope")
figure_LMM_GER_ITA$corr_uncorrected$ALL
figure_LMM_GER_ITA$corr_corrected$ALL



### Use partial data to calculate multifunctionality by all plots in country, not by pairs
### (Take the first 3 plots in each country)
```

```
data("forest_function_data_raw")
data("forest_biodiversity_data")

forest_function_data_raw_3plots <- forest_function_data_raw[c(1:3,29:31,67:69,103:105,
                                                              146:148,174:176),]
forest_function_data_normalized_3plots <-
                    function_normalization(data = forest_function_data_raw_3plots,
                    fun_cols = 6:31,
                    negative = c("soil_cn_ff_10","wue"),
                    by_group = "country")
forest_biodiversity_data_3plots <-
                forest_biodiversity_data[c(1:6,49:52,141:148,230:232,351:355,411:417),]

output5 <- MF2_multiple(func_data = forest_function_data_normalized_3plots[,6:32],
                        species_data = forest_biodiversity_data_3plots,
                        weight = 1,
                        by_group = "country", by_pair = FALSE)

## Display fitted line of linear model
figure_all_plots <- MFggplot(output5, model = "lm", caption = "slope")
figure_all_plots$corr_uncorrected$ALL
figure_all_plots$corr_corrected$ALL
```

Index