Package 'MicrobiotaProcess'

January 24, 2026

Type Package

Title A comprehensive R package for managing and analyzing microbiome and other ecological data within the tidy framework

Version 1.23.0

Description MicrobiotaProcess is an R package for analysis, visualization and biomarker discovery of microbial datasets. It introduces MPSE class, this make it more interoperable with the existing computing ecosystem. Moreover, it introduces a tidy microbiome data structure paradigm and analysis grammar. It provides a wide variety of microbiome data analysis procedures under the unified and common framework (tidy-like framework).

Depends R (>= 4.0.0)

Imports ape, tidyr, ggplot2, magrittr, dplyr, Biostrings, ggrepel, vegan, zoo, ggtree, tidytree (>= 0.4.2), MASS, methods, rlang, tibble, grDevices, stats, utils, coin, ggsignif, patchwork, ggstar, tidyselect, SummarizedExperiment, foreach, treeio (>= 1.17.2), pillar, cli, plyr, dtplyr, ggtreeExtra, data.table, ggfun (>= 0.1.1)

Suggests rmarkdown, prettydoc, testthat, knitr, nlme, phangorn, DECIPHER, randomForest, jsonlite, biomformat, scales, yaml, withr, S4Vectors, purrr, seqmagick, glue, ggupset, ggVennDiagram, ggalluvial (>= 0.11.1), forcats, phyloseq, aplot, ggnewscale, ggside, ggh4x, hopach, parallel, shadowtext, DirichletMultinomial, ggpp, BiocManager

License GPL (>= 3.0)

URL https://github.com/YuLab-SMU/MicrobiotaProcess/

BugReports https://github.com/YuLab-SMU/MicrobiotaProcess/issues

VignetteBuilder knitr

ByteCompile true

Encoding UTF-8

biocViews Visualization, Microbiome, Software, MultipleComparison, FeatureExtraction

RoxygenNote 7.3.2

git_url https://git.bioconductor.org/packages/MicrobiotaProcess

git_branch devel

git_last_commit 9b6279e

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2026-01-23

Author Shuangbin Xu [aut, cre] (ORCID:
<https://orcid.org/0000-0003-3513-5362>),
Guangchuang Yu [aut, ctb] (ORCID:
<https://orcid.org/0000-0002-6485-8781>)

Maintainer Shuangbin Xu <xshuangbin@163.com>

Contents

alphasample-class *alphasample class*

Description

alphasample class

Slots

`alpha` data.frame contained alpha metrics of samples

`sampleda` associated sample information

as.MPSE *as.MPSE method*

Description

convert the .data object to MPSE object

Usage

```
as.MPSE(.data, ...)

as.mpse(.data, ...)
```

Arguments

`.data`	one type of tbl_mpse, phyloseq, biom, SummarizedExperiment or TreeSummarizedExperiment class
`...`	additional parameters, meaningless now.

Value

MPSE object

Author(s)

Shuangbin Xu

as.phyloseq *convert to phyloseq object.*

Description

convert to phyloseq object.

Usage

```
as.phyloseq(x, .abundance, ...)

as_phyloseq(x, .abundance, ...)

## S3 method for class 'MPSE'
as.phyloseq(x, .abundance, ...)

## S3 method for class 'tbl_mpse'
as.phyloseq(x, .abundance, ...)
```

Arguments

x	object, tbl_mpse object, which the result of as_tibble for phyloseq objcet.
.abundance	the column name to be as the abundance of otu table, default is Abundance.
...	additional params

Value

phyloseq object.

as.treedata.taxonomyTable *as.treedata*

Description

convert taxonomyTable to treedata

Usage

```
## S3 method for class 'taxonomyTable'
as.treedata(tree, include.rownames = FALSE, ...)
```

Arguments

tree	object, This is for taxonomyTable class, so it should be a taxonomyTable.
include.rownames	logical, whether to set the rownames of taxonomyTable to tip labels, default is FALSE.
...	additional parameters.

Examples

```
## Not run:
  data(test_otu_data)
  test_otu_data %<>% as.phyloseq()
  tree <- as.treedata(phyloseq::tax_table(test_otu_data), include.rownames = TRUE)

## End(Not run)
```

---

build_tree *building tree*

Description

The function can be used to building tree.

Usage

```
build_tree(seqs, ...)

## S4 method for signature 'DNAStringSet'
build_tree(seqs, ...)

## S4 method for signature 'DNAbin'
build_tree(seqs, ...)

## S4 method for signature 'character'
build_tree(seqs, ...)
```

Arguments

seqs	DNAStringSet or DNAbin, the object of R.
...	additional parameters, see also AlignSeqs.

Value

the phylo class of tree.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
    seqtabfile <- system.file("extdata", "seqtab.nochim.rds",
                              package="MicrobiotaProcess")
    seqtab <- readRDS(seqtabfile)
    refseq <- colnames(seqtab)
    names(refseq) <- paste0("OTU_",seq_len(length(refseq)))
    refseq <- Biostrings::DNAStringSet(refseq)
    tree <- build_tree(refseq)
    or
    tree <- build_tree(refseq)

## End(Not run)
```

`convert_to_treedata` *convert dataframe contained hierarchical relationship or other classes to treedata class*

Description

convert dataframe contained hierarchical relationship or other classes to treedata class

Usage

```
convert_to_treedata(data, type = "species", include.rownames = FALSE, ...)
```

Arguments

`data`	data.frame, such like the tax_table of phyloseq.
`type`	character, the type of datasets, default is "species", if the dataset is not about species, #' such as dataset of kegg function, you should set it to "others".
`include.rownames`	logical, whether to set the row names as the tip labels, default is FALSE.
`...`	additional parameters.

Value

treedata class.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
  data(hmp_aerobiosis_small)
  head(taxda)
  treedat <- convert_to_treedata(taxda, include.rownames = FALSE)

## End(Not run)
```

`data-hmp_aerobiosis_small` *(Data) Small subset of the HMP 16S dataset*

Description

Contained three datasets, featureda, sampleda, taxda featureda contained 55 samples (nrow) and 1091 features (ncol) sampleda contained 55 samples from 6 body sites of 10 subjects. taxda contained 699 taxonomy by 6 rank. This datasets were built from the LEfSe.http://huttenhower.sph.harvard.edu/webfm_send/129

Examples

```
data(hmp_aerobiosis_small)
```

`data-kostic2012crc` *(Data) Genomic analysis identifies association of Fusobacterium with colorectal carcinoma (2012)*

Description

This dataset was from the a study on colorectal cancer, publised in Genome Research (2012). This dataset had been removed samples with less than 500 reads, contained 91 Control and 86 Tumors. And It is belong to MPSE class, contained otu_table and sample_data.

Examples

```
data(kostic2012crc)
```

`data-test_otu_data` *(Data) simulated dataset.*

Description

This dataset was simulated. And it also was MPSE class, contained otu_table and sample_data

Examples

```
data(test_otu_data)
```

`diffAnalysisClass-class` *diffAnalysisClass class*

Description

diffAnalysisClass class

Slots

`originalD` original feature data.frame.

`sampleda` associated sample information.

`taxda` the data.frame contained taxonomy.

`result` data.frame contained the results of first, second test and LDA or rf

`kwres` the results of first test, contained feature names, pvalue and fdr.

`secondvars` the results of second test, contained features names, gfc (TRUE representation the relevant feantures is enriched in relevant factorNames), Freq(the number of TRUE or FALSE), factorNames.

`mlres` the results of LDA or randomForest,

`someparams,` some arguments will be used in other functions `diff_analysis`

`diff_analysis` *Differential expression analysis*

Description

Differential expression analysis

Usage

```
diff_analysis(obj, ...)

## S3 method for class 'data.frame'
diff_analysis(
  obj,
  sampleda,
  classgroup,
  subclass = NULL,
  taxda = NULL,
  alltax = TRUE,
  include.rownames = FALSE,
  standard_method = NULL,
  mlfun = "lda",
```

```
  ratio = 0.7,
  firstcomfun = "kruskal.test",
  padjust = "fdr",
  filtermod = "pvalue",
  firstalpha = 0.05,
  strictmod = TRUE,
  fcfun = "generalizedFC",
  secondcomfun = "wilcox.test",
  clmin = 5,
  clwilc = TRUE,
  secondalpha = 0.05,
  subclmin = 3,
  subclwilc = TRUE,
  ldascore = 2,
  normalization = 1e+06,
  bootnums = 30,
  ci = 0.95,
  type = "species",
  ...
)

## S3 method for class 'phyloseq'
diff_analysis(obj, ...)
```

Arguments

obj	object,a phyloseq class contained otu_table, sample_data, taxda, or data.frame, nrow sample * ncol features.
...	additional parameters.
sampleda	data.frame, nrow sample * ncol factor, the sample names of sampleda and data should be the same.
classgroup	character, the factor name in sampleda.
subclass	character, the factor name in sampleda, default is NULL, meaning no subclass compare.
taxda	data.frame, the classification of the feature in data. default is NULL.
alltax	logical, whether to set all classification (taxonomy) as features when taxda is not NULL, default is TRUE.
include.rownames	logical, whether to consider the OTU of obj as (all taxonomy) features, when taxda is not NULL, default is FALSE.
standard_method	character, the method of standardization, see also decostand, default is NULL, it represents that the relative abundance of taxonomy will be used. If count was set, it represents the count reads of taxonomy will be used.
mlfun	character, the method for calculating the effect size of features, choose "lda" or "rf", default is "lda".

ratio	numeric, range from 0 to 1, the proportion of samples for calculating the effect size of features, default is 0.7.
firstcomfun	character, the method for first test, "oneway.test" for normal distributions, suggested choosing "kruskal.test" for uneven distributions, default is "kruskal.test", or you can use lm, glm, or glm.nb (for negative binomial distribution), or 'kruskal_test', 'oneway_test' of 'coin'.
padjust	character, the correction method, default is "fdr".
filtermod	character, the method to filter, default is "pvalue".
firstalpha	numeric, the alpha value for the first test, default is 0.05.
strictmod	logical, whether to performed in one-against-one, default is TRUE (strict).
fcfun	character, default is "generalizedFC", it can't be set another at the present time.
secondcomfun	character, the method for one-against-one, default is "wilcox.test" for uneven distributions, or 'wilcox_test' of 'coin', or you can also use 'lm', 'glm', 'glm.nb'(for negative binomial distribution in 'MASS').
clmin	integer, the minimum number of samples per classgroup for performing test, default is 5.
clwilc	logical, whether to perform test of per classgroup, default is TRUE.
secondalpha	numeric, the alpha value for the second test, default is 0.05.
subclmin	integer, the minimum number of samples per suclass for performing test, default is 3.
subclwilc	logical, whether to perform test of per subclass, default is TRUE, meaning more strict.
ldascore	numeric, the threshold on the absolute value of the logarithmic LDA score, default is 2.
normalization	integer, set the normalization value, set a big number if to get more meaningful values for the LDA score, or you can set NULL for no normalization, default is 1000000.
bootnums	integer, set the number of bootstrap iteration for lda or rf, default is 30.
ci	numeric, the confidence interval of effect size (LDA or MDA), default is 0.95.
type	character, the type of datasets, default is "species", if the dataset is not about species, such as dataset of kegg function, you should set it to "others".

Value

diff_analysis class.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(kostic2012crc)
kostic2012crc %<>% as.phyloseq()
head(phyloseq::sample_data(kostic2012crc),3)
kostic2012crc <- phyloseq::rarefy_even_depth(kostic2012crc,rngseed=1024)
table(phyloseq::sample_data(kostic2012crc)$DIAGNOSIS)
set.seed(1024)
diffres <- diff_analysis(kostic2012crc, classgroup="DIAGNOSIS",
                         mlfun="lda", filtermod="fdr",
                         firstcomfun = "kruskal.test",
                         firstalpha=0.05, strictmod=TRUE,
                         secondcomfun = "wilcox.test",
                         subclmin=3, subclwilc=TRUE,
                         secondalpha=0.01, ldascore=3)

## End(Not run)
```

---

drop_taxa *Dropping Species with Few abundance and Few Occurrences*

Description

Drop species or features from the feature data frame or phyloseq that occur fewer than or equal to a threshold number of occurrences and fewer abundance than to a threshold abundance.

Usage

```
drop_taxa(obj, ...)

## S4 method for signature 'data.frame'
drop_taxa(obj, minocc = 0, minabu = 0, ...)

## S4 method for signature 'phyloseq'
drop_taxa(obj, ...)
```

Arguments

obj	object, phyloseq or a dataframe of species (n_sample, n_feature).
...	additional parameters.
minocc	numeric, the threshold number of occurrences to be dropped, if < 1.0,it will be the threshold ratios of occurrences, default is 0.
minabu	numeric, the threshold abundance, if fewer than the threshold will be dropped, default is 0.

Value

dataframe of new features.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
otudafile <- system.file("extdata", "otu_tax_table.txt",
                         package="MicrobiotaProcess")
otuda <- read.table(otudafile, sep="\t",
                    header=TRUE, row.names=1,
                    check.names=FALSE, skip=1,
                    comment.char="")
otuda <- otuda[sapply(otuda, is.numeric)]
otuda <- data.frame(t(otuda), check.names=FALSE)
dim(otuda)
otudat <- drop_taxa(otuda, minocc=0.1, minabu=1)
dim(otudat)
data(test_otu_data)
test_otu_data %<>% as.phyloseq()
keepps <- drop_taxa(test_otu_data, minocc=0.1, minabu=0)

## End(Not run)
```

---

dr_extract *Extracting the internal tbl_df attribute of tibble.*

Description

Extracting the internal tbl_df attribute of tibble.

Usage

```
dr_extract(name, .f = NULL)
```

Arguments

name	character the name of internal tbl_df attribute.
.f	a function (if any, default is NULL) that pre-operate the data

Value

tbl_df object

Author(s)

Shuangbin Xu

Examples

```
## Not run:
library(vegan)
data(varespec, varechem)
mpse <- MPSE(assays=list(Abundance=t(varespec)), colData=varechem)
tbl <-
mpse %>%
  mp_cal_nmds(.abundance=Abundance, action="add") %>%
  mp_envfit(.ord=NMDS, .env=colnames(varechem), action="only")
tbl
tbl %>% attributes %>% names
# This function is useful to extract the data to display with ggplot2
# you can also refer to the examples of mp_envfit.
dr_extract(name=NMDS_ENVFIT_tb)(tbl)
# add .f function
dr_extract(name=NMDS_ENVFIT_tb,
           .f=td_filter(pvals<=0.05 & label!="Humdepth"))(tbl)

## End(Not run)
```

---

extract_binary_offspring

extract the binary offspring of the specified internal nodes

Description

extract the binary offspring of the specified internal nodes

Usage

```
extract_binary_offspring(.data, .node, type = "tips", ...)
```

Arguments

.data	phylo or treedata object
.node	the internal nodes
type	the type of binary offspring, options are 'tips' (default), 'all', 'internal'.
...	additional parameter, meaningless now.

generalizedFC *generalized fold change*

Description

calculate the mean difference in a set of predefined quantiles of the logarithmic

Usage

```
generalizedFC(x, ...)

## Default S3 method:
generalizedFC(x, y, base = 10, steps = 0.05, pseudo = 1e-05, ...)

## S3 method for class 'formula'
generalizedFC(x, data, subset, na.action, ...)
```

Arguments

x	numeric vector, numeric vector of data values or formula, example 'Ozone ~ Month', Ozone is a numeric variable giving the data values 'Month' a factor giving the corresponding groups.
...	additional arguments.
y	numeric vector, numeric vector of data values
base	a positive or complex number, the base with respect to which logarithms are computed, default is 10.
steps	positive numeric, increment of the sequence, default is 0.05.
pseudo	positive numeric, avoid the zero for logarithmic, default is 0.00001.
data	data.frame, an optional matrix or data frame,containing the variables in the formula.
subset	(similar: see 'wilcox.test')an optional vector specifying a subset of observations to be used.
na.action	a function which indicates what should happen when the data, contain 'NA's. Defaults to 'getOption("na.action")'.

Value

list contained gfc, the mean and median of different group.

Author(s)

Shuangbin Xu

Examples

```
set.seed(1024)
data <- data.frame(A=rnorm(1:10,mean=5),
                   B=rnorm(2:11, mean=6),
                   group=c(rep("case",5),rep("control",5)))
generalizedFC(B ~ group,data=data)
generalizedFC(x=c(1,2,3,4,5),y=c(3,4,5,6,7))
```

`get_alltaxadf` *get the table of abundance of all level taxonomy*

Description

This function was designed to get the abundance of all level taxonomy, the input can be phyloseq object or data.frame.

Usage

```
get_alltaxadf(obj, ...)

## S4 method for signature 'phyloseq'
get_alltaxadf(
  obj,
  method = NULL,
  type = "species",
  include.rownames = FALSE,
  ...
)

## S4 method for signature 'data.frame'
get_alltaxadf(
  obj,
  taxda,
  taxa_are_rows = FALSE,
  method = NULL,
  type = "species",
  include.rownames = FALSE,
  ...
)
```

Arguments

`obj`	object, phyloseq or data.frame
`...`	additional parameters, see also `decostand`.
`method`	character, the normalization method, see also `decostand`, default is NULL, the relative abundance will be return, if it set 'count', the count table will be return.

include.rownames
: logical whether to calculate the original feature data, default is FALSE.

taxda
: data.frame, the taxonomy table.

taxa_are_rows
: logical, if the obj is data.frame, and the features are rownames, the taxa_are_rows should be set TRUE, default FALSE, meaning the features are colnames.

Value

the all taxonomy abundance table

Author(s)

Shuangbin Xu

Examples

```
## Not run:
   data(test_otu_data)
   alltaxatab <- get_alltaxadf(test_otu_data)
   head(alltaxatab[,1:10])

## End(Not run)
```

---

get_alphaindex *alpha index*

Description

calculate the alpha index (Obseve,Chao1,Shannon,Simpson) of sample with diversity

Usage

```
get_alphaindex(obj, ...)

## S4 method for signature 'matrix'
get_alphaindex(obj, mindepth, sampleda, force = FALSE, ...)

## S4 method for signature 'data.frame'
get_alphaindex(obj, ...)

## S4 method for signature 'integer'
get_alphaindex(obj, ...)

## S4 method for signature 'numeric'
get_alphaindex(obj, ...)

## S4 method for signature 'phyloseq'
get_alphaindex(obj, ...)
```

Arguments

obj	object, data.frame of (nrow sample * ncol taxonomy(feature)) or phyloseq.
...	additional arguments.
mindepth	numeric, Subsample size for rarefying community.
sampleda	data.frame,sample information, row sample * column factors.
force	logical whether calculate the alpha index even the count of otu is not rarefied, default is FALSE. If it is TRUE, meaning the rarefaction is not be performed automatically.

Value

data.frame contained alpha Index.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
otudafile <- system.file("extdata", "otu_tax_table.txt",
                         package="MicrobiotaProcess")
otuda <- read.table(otudafile, sep="\t",
             header=TRUE, row.names=1,
             check.names=FALSE, skip=1, comment.char="")
otuda <- otuda[sapply(otuda, is.numeric)] %>% t() %>%
          data.frame(check.names=FALSE)
set.seed(1024)
alphatab <- get_alphaindex(otuda)
head(as.data.frame(alphatab))
data(test_otu_data)
class(test_otu_data)
test_otu_data %<>% as.phyloseq()
class(test_otu_data)
set.seed(1024)
alphatab2 <- get_alphaindex(test_otu_data)
head(as.data.frame(alphatab2))

## End(Not run)
```

---

get_clust *Hierarchical cluster analysis for the samples*

Description

Hierarchical cluster analysis for the samples

Usage

```
get_clust(obj, ...)

## S3 method for class 'dist'
get_clust(obj, distmethod, sampleda = NULL, hclustmethod = "average", ...)

## S3 method for class 'data.frame'
get_clust(
  obj,
  distmethod = "euclidean",
  taxa_are_rows = FALSE,
  sampleda = NULL,
  tree = NULL,
  method = "hellinger",
  hclustmethod = "average",
  ...
)

## S3 method for class 'phyloseq'
get_clust(
  obj,
  distmethod = "euclidean",
  method = "hellinger",
  hclustmethod = "average",
  ...
)
```

Arguments

obj	phyloseq, phyloseq class or dist class, or data.frame, data.frame, default is nrow samples * ncol features.
...	additional parameters.
distmethod	character, the method of dist, when the obj is data.frame or phyloseq default is "euclidean". see also get_dist.
sampleda	data.frame, nrow sample * ncol factor. default is NULL.
hclustmethod	character, the method of hierarchical cluster, default is average.
taxa_are_rows	logical, if the features of data.frame(obj) is in column, it should set FALSE.
tree	phylo, the phylo class, see also as.phylo.
method	character, the standardization methods for community ecologists, see also decostand

Value

treedata object.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
library(phyloseq)
data(GlobalPatterns)
subGlobal <- subset_samples(GlobalPatterns,
         SampleType %in% c("Feces", "Mock", "Ocean", "Skin"))
hcsample <- get_clust(subGlobal, distmethod="jaccard",
                  method="hellinger", hclustmethod="average")

## End(Not run)
```

`get_coord.pcoa` *get ordination coordinates.*

Description

get ordination coordinates.

Usage

```
## S3 method for class 'pcoa'
get_coord(obj, pc)

get_coord(obj, pc)

## S3 method for class 'prcomp'
get_coord(obj, pc)
```

Arguments

`obj`	object,prcomp class or pcoa class
`pc`	integer vector, the component index.

Value

ordplotClass object.

Examples

```
## Not run:
require(graphics)
data(USArrests)
pcares <- prcomp(USArrests, scale = TRUE)
coordtab <- get_coord(pcares,pc=c(1, 2))
coordtab2 <- get_coord(pcares, pc=c(2, 3))

## End(Not run)
```

get_count *calculate the count or relative abundance of replicate element with a speficify column*

Description

Caculate the count or relative abundance of replicate element with a speficify columns

Usage

```
get_count(data, featurelist, ...)

get_ratio(data, featurelist, ...)
```

Arguments

data	dataframe; a dataframe contained one character column and others is numeric, if featurelist is NULL. Or a numeirc dataframe, if featurelist is non't NULL, all columns should be numeric.
featurelist	dataframe; a dataframe contained one chatacter column, default is NULL.
...	additional parameters.

Value

mean of data.frame by featurelist

Author(s)

Shuangbin Xu

Examples

```
## Not run:
otudafile <- system.file("extdata", "otu_tax_table.txt",
                      package="MicrobiotaProcess")
samplefile <- system.file("extdata",
                "sample_info.txt", package="MicrobiotaProcess")
otuda <- read.table(otudafile, sep="\t", header=TRUE,
                    row.names=1, check.names=FALSE,
                    skip=1, comment.char="")
sampleda <- read.table(samplefile,
            sep="\t", header=TRUE, row.names=1)
taxdf <- otuda[!sapply(otuda, is.numeric)]
taxdf <- split_str_to_list(taxdf)
otuda <- otuda[sapply(otuda, is.numeric)]
phycount <- get_count(otuda, taxdf[,2,drop=FALSE])
phyratios <- get_ratio(otuda, taxdf[,2,drop=FALSE])

## End(Not run)
```

get_dist *calculate distance*

Description

calculate distance

Usage

```
get_dist(obj, ...)

## S3 method for class 'data.frame'
get_dist(
  obj,
  distmethod = "euclidean",
  taxa_are_rows = FALSE,
  sampleda = NULL,
  tree = NULL,
  method = "hellinger",
  ...
)

## S3 method for class 'phyloseq'
get_dist(obj, distmethod = "euclidean", method = "hellinger", ...)
```

Arguments

obj	phyloseq, phyloseq class or data.frame nrow sample * ncol feature.
...	additional parameters.
distmethod	character, default is "euclidean", see also distanceMethodList
taxa_are_rows	logical, default is FALSE.
sampleda	data.frame, nrow sample * ncol factors.
tree	object, the phylo class, see also as.phylo.
method	character, default is hellinger, see alse decostand

Value

distance class contianed distmethod and originalD attr

See Also

distance

Examples

```
## Not run:
data(test_otu_data)
test_otu_data %<>% as.phyloseq()
distclass <- get_dist(test_otu_data)
hcsample <- get_clust(distclass)

## End(Not run)
```

`get_mean_median` *get the mean and median of specific feature.*

Description

get the mean and median of specific feature.

Usage

```
get_mean_median(datameta, feature, subclass)
```

Arguments

`datameta`	data.frame, nrow sample * ncol feature + factor.
`feature`	character vector, the feature contained in datameta.
`subclass`	character, factor name.

Value

featureMeanMedian object, contained the abundance of feature, and the mean and median of feature by subclass.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
    data(hmp_aerobiosis_small)
    head(sampleda)
    featureda <- merge(featureda, sampleda, by=0)
    rownames(featureda) <- as.vector(featureda$Row.names)
    featureda$Row.names <- NULL
    feameamed <- get_mean_median(datameta=featureda,
                        feature="p__Actinobacteria",
                        subclass="body_site")
    fplot <- ggdifftaxbar(feameamed, featurename="p__Actinobacteria",
                         classgroup="oxygen_availability", subclass="body_site")

## End(Not run)
```

get_NRI_NTI *calculating related phylogenetic alpha metric*

Description

calculating related phylogenetic alpha metric

Usage

```
get_NRI_NTI(obj, ...)

## S4 method for signature 'matrix'
get_NRI_NTI(
  obj,
  mindepth,
  sampleda,
  tree,
  metric = c("PAE", "NRI", "NTI", "PD", "HAED", "EAED", "IAC", "all"),
  abundance.weighted = FALSE,
  force = FALSE,
  seed = 123,
  ...
)

## S4 method for signature 'data.frame'
get_NRI_NTI(obj, mindepth, sampleda, tree, abundance.weighted = TRUE, ...)

## S4 method for signature 'phyloseq'
get_NRI_NTI(obj, mindepth, abundance.weighted = TRUE, ...)
```

Arguments

obj	object, data.frame of (nrow sample * ncol taxonomy(feature)) or phyloseq.
...	additional arguments, meaningless now.
mindepth	numeric, Subsample size for rarefying community.
sampleda	data.frame, sample information, row sample * column factors.
tree	tree object, it can be phylo object or treedata object.
metric	the related phylogenetic metric, options is 'NRI', 'NTI', 'PD', 'PAE', 'HAED', 'EAED', 'IAC', 'all', default is 'PAE', meaning all the metrics ('NRI', 'NTI', 'PD', 'PAE', 'HAED', 'EAED', 'IAC').
abundance.weighted	logical, whether calculate mean nearest taxon distances for each species weighted by species abundance, default is FALSE.
force	logical whether calculate the index even the count of otu is not rarefied, default is FALSE. If it is TRUE, meaning the rarefaction is not be performed automatically.

seed	integer a random seed to make the result reproducible, default is 123.

Value

alphasample object contained NRT and NTI.

Author(s)

Shuangbin Xu

`get_pca` *Performs a principal components analysis*

Description

Performs a principal components analysis

Usage

```
get_pca(obj, ...)

## S3 method for class 'data.frame'
get_pca(obj, sampleda = NULL, method = "hellinger", ...)

## S3 method for class 'phyloseq'
get_pca(obj, method = "hellinger", ...)
```

Arguments

obj	phyloseq, phyloseq class or data.frame shape of data.frame is nrow sample * ncol feature.
...	additional parameters, seeprcomp.
sampleda	data.frame, nrow sample * ncol factors.
method	character, the standardization methods for community ecologists. see decostand.

Value

pcasample class, contained prcomp class and sample information.

Examples

```
## Not run:
library(phyloseq)
data(GlobalPatterns)
subGlobal <- subset_samples(GlobalPatterns,
         SampleType %in% c("Feces", "Mock", "Ocean", "Skin"))
pcares <- get_pca(subGlobal, method="hellinger")
pcaplot <- ggordpoint(pcares, biplot=TRUE,
```

```
                    speciesannot=TRUE,
                    factorNames=c("SampleType"), ellipse=TRUE)

## End(Not run)
```

`get_pcoa` *performs principal coordinate analysis (PCoA)*

Description

performs principal coordinate analysis (PCoA)

Usage

```
get_pcoa(obj, ...)

## S3 method for class 'data.frame'
get_pcoa(
  obj,
  distmethod = "euclidean",
  taxa_are_rows = FALSE,
  sampleda = NULL,
  tree = NULL,
  method = "hellinger",
  ...
)

## S3 method for class 'dist'
get_pcoa(
  obj,
  distmethod,
  data = NULL,
  sampleda = NULL,
  method = "hellinger",
  ...
)

## S3 method for class 'phyloseq'
get_pcoa(obj, distmethod = "euclidean", ...)
```

Arguments

`obj`	phyloseq, the phyloseq class or dist class.
`...`	additional parameter, see also `get_dist`.
`distmethod`	character, the method of distance, see also `distance`
`taxa_are_rows`	logical, if feature of data is column, it should be set FALSE.

sampleda	data.frame, nrow sample * ncol factor, default is NULL.
tree	phylo, the phylo class, default is NULL, when use unifrac method, it should be required.
method	character, the standardization method for community ecologists, default is hellinger, if the data has be normlized, it shound be set NULL.
data	data.frame, numeric data.frame nrow sample * ncol features.

Value

pcasample object, contained prcomp or pcoa and sampleda (data.frame).

Author(s)

Shuangbin Xu

Examples

```
## Not run:
    library(phyloseq)
    data(GlobalPatterns)
    subGlobal <- subset_samples(GlobalPatterns,
                  SampleType %in% c("Feces", "Mock", "Ocean", "Skin"))
    pcoares <- get_pcoa(subGlobal,
                        distmethod="euclidean",
                        method="hellinger")
    pcoaplot <- ggordpoint(pcoares, biplot=FALSE,
                           speciesannot=FALSE,
                           factorNames=c("SampleType"),
                           ellipse=FALSE)

## End(Not run)
```

---

get_pvalue *Methods for computation of the p-value*

Description

Methods for computation of the p-value

Usage

```
get_pvalue(obj)

## S3 method for class 'htest'
get_pvalue(obj)

## S3 method for class 'lme'
get_pvalue(obj)
```

```
## S3 method for class 'negbin'
get_pvalue(obj)

## S3 method for class 'ScalarIndependenceTest'
get_pvalue(obj)

## S3 method for class 'QuadTypeIndependenceTest'
get_pvalue(obj)

## S3 method for class 'lm'
get_pvalue(obj)

## S3 method for class 'glm'
get_pvalue(obj)
```

Arguments

obj	object, such as htest, lm, negbin ScalarIndependenceTest class.

Value

pvalue.

Author(s)

Shuangbin Xu

Examples

```
library(nlme)
lmeres <- lme(distance ~ Sex,data=Orthodont)
pvalue <- get_pvalue(lmeres)
```

---

get_rarecurve *obtain the result of rare curve*

Description

generate the result of rare curve.

Usage

```
get_rarecurve(obj, ...)

## S4 method for signature 'data.frame'
get_rarecurve(obj, sampleda, factorLevels = NULL, chunks = 400)

## S4 method for signature 'phyloseq'
get_rarecurve(obj, ...)
```

Arguments

obj	phyloseq class or data.frame shape of data.frame (nrow sample * ncol feature)
...	additional parameters.
sampleda	data.frame, (nrow sample * ncol factor)
factorLevels	list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.
chunks	integer, the number of subsample in a sample, default is 400.

Details

This function is designed to calculate the rare curve result of otu table the result can be visualized by ‘ggrarecurve‘.

Value

rarecurve class, which can be visualized by ggrarecurve

Author(s)

Shuangbin Xu

Examples

```
## Not run:
    data(test_otu_data)
    test_otu_data %<>% as.phyloseq()
    set.seed(1024)
    res <- get_rarecurve(test_otu_data, chunks=200)
    p <- ggrarecurve(obj=res,
                     indexNames=c("Observe","Chao1","ACE"),
                     shadow=FALSE,
                     factorNames="group")

## End(Not run)
```

---

get_sampledflist *Generate random data list from a original data.*

Description

Generate random data list from a original data.

Usage

```
get_sampledflist(dalist, bootnums = 30, ratio = 0.7, makerownames = FALSE)
```

Arguments

dalist	list, a list contained multi data.frame.
bootnums	integer, the number of bootstrap iteration, default is 30.
ratio	numeric, the ratios of each data.frame to keep.
makerownames	logical, whether build row.names,default is FALSE.

Value

the list contained the data.frame generated by bootstrap iteration.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
    data(iris)
    irislist <- split(iris, iris$Species)
    set.seed(1024)
    irislist <- get_sampledflist(irislist)

## End(Not run)
```

---

get_taxadf *get the data of specified taxonomy*

Description

get the data of specified taxonomy

Usage

```
get_taxadf(obj, ...)

## S4 method for signature 'phyloseq'
get_taxadf(obj, taxlevel = 2, type = "species", ...)

## S4 method for signature 'data.frame'
get_taxadf(
  obj,
  taxda,
  taxa_are_rows,
  taxlevel,
  sampleda = NULL,
  type = "species",
  ...
)
```

Arguments

obj	phyloseq, phyloseq class or data.frame the shape of data.frame (nrow sample * column feature taxa_are_rows set FALSE, nrow feature * ncol sample, taxa_are_rows set TRUE).
...	additional parameters.
taxlevel	character, the column names of taxda that you want to get. when the input is phyloseq class, you can use 1 to 7.
type	character, the type of datasets, default is "species", if the dataset is not about species, such as dataset of kegg function, you should set it to "others".
taxda	data.frame, the classifies of feature contained in obj(data.frame).
taxa_are_rows	logical, if the column of data.frame are features, it should be set FALSE.
sampleda	data.frame, the sample information.

Value

phyloseq class contained tax data.frame and sample information.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
library(ggplot2)
data(test_otu_data)
test_otu_data %<>% as.phyloseq()
phytax <- get_taxadf(test_otu_data, taxlevel=2)
phytax
head(phyloseq::otu_table(phytax))
phybar <- ggbartax(phytax) +
         xlab(NULL) + ylab("relative abundance (%)")

## End(Not run)
```

---

get_upset *generate the dataset for upset of UpSetR*

Description

generate the dataset for upset of UpSetR

Usage

```
get_upset(obj, ...)

## S4 method for signature 'data.frame'
get_upset(obj, sampleda, factorNames, threshold = 0)

## S4 method for signature 'phyloseq'
get_upset(obj, ...)
```

Arguments

obj	object, phyloseq or data.frame, if it is data.frame, the shape of it should be row sample * columns features.
...	additional parameters.
sampleda	data.frame, if the obj is data.frame, the sampleda should be provided.
factorNames	character, the column names of factor in sampleda
threshold	integer, default is 0.

Value

a data.frame for the input of ‘upset‘ of ‘UpSetR‘.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(test_otu_data)
test_otu_data %<>% as.phyloseq()
upsetda <- get_upset(test_otu_data, factorNames="group")
otudafile <- system.file("extdata", "otu_tax_table.txt",
                         package="MicrobiotaProcess")
samplefile <- system.file("extdata","sample_info.txt",
                         package="MicrobiotaProcess")
otuda <- read.table(otudafile, sep="\t", header=TRUE,
                    row.names=1, check.names=FALSE,
                    skip=1, comment.char="")
sampleda <- read.table(samplefile,sep="\t",
                       header=TRUE, row.names=1)
head(sampleda)
otuda <- otuda[sapply(otuda, is.numeric)]
otuda <- data.frame(t(otuda), check.names=FALSE)
head(otuda[1:5, 1:5])
upsetda2 <- get_upset(obj=otuda, sampleda=sampleda,
                      factorNames="group")
#Then you can use `upset` of `UpSetR` to visualize the results.
library(UpSetR)
```

```
upset(upsetda, sets=c("B","D","M","N"), sets.bar.color = "#56B4E9",
      order.by = "freq", empty.intersections = "on")

## End(Not run)
```

`get_varct.pcoa` *get the contribution of variables*

Description

get the contribution of variables

Usage

```
## S3 method for class 'pcoa'
get_varct(obj, ...)

get_varct(obj, ...)

## S3 method for class 'prcomp'
get_varct(obj, ...)

## S3 method for class 'pcasample'
get_varct(obj, ...)
```

Arguments

`obj`	prcomp class or pcasample class
`...`	additional parameters.

Value

the VarContrib class, contained the contribution and coordinate of features.

Examples

```
## Not run:
library(phyloseq)
data(GlobalPatterns)
subGlobal <- subset_samples(GlobalPatterns,
         SampleType %in% c("Feces", "Mock", "Ocean", "Skin"))
pcares <- get_pca(subGlobal, method="hellinger")
varres <- get_varct(pcares)

## End(Not run)
```

get_vennlist *generate a vennlist for VennDiagram*

Description

generate a vennlist for VennDiagram

Usage

```
get_vennlist(obj, ...)

## S4 method for signature 'phyloseq'
get_vennlist(obj, factorNames, ...)

## S4 method for signature 'data.frame'
get_vennlist(obj, sampleinfo = NULL, factorNames = NULL, ...)
```

Arguments

obj	phyloseq, phyloseq class or data.frame a dataframe contained one character column and the others are numeric. or all columns should be numeric if sampleinfo isn't NULL.
...	additional parameters
factorNames	character, a column name of sampleinfo, when sampleinfo isn't NULL, factorNames shouldn't be NULL, default is NULL, when the input is phyloseq, the factorNames should be provided.
sampleinfo	dataframe; a sample information, default is NULL.

Value

return a list for VennDiagram.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(test_otu_data)
test_otu_data %<>% as.phyloseq()
vennlist <- get_vennlist(test_otu_data,
                 factorNames="group")
vennlist
library(VennDiagram)
venn.diagram(vennlist, height=5,
             width=5, filename = "./test_venn.pdf",
```

```
              alpha = 0.85, fontfamily = "serif",
              fontface = "bold",cex = 1.2,
              cat.cex = 1.2, cat.default.pos = "outer",
              cat.dist = c(0.22,0.22,0.12,0.12),
              margin = 0.1, lwd = 3,
              lty ='dotted',
              imagetype = "pdf")

## End(Not run)
```

---

ggbartax *taxonomy barplot*

Description

taxonomy barplot

Usage

```
ggbartax(obj, ...)

ggbartaxa(obj, ...)

## S3 method for class 'phyloseq'
ggbartax(obj, ...)

## S3 method for class 'data.frame'
ggbartax(
  obj,
  mapping = NULL,
  position = "stack",
  stat = "identity",
  width = 0.7,
  topn = 30,
  count = FALSE,
  sampleda = NULL,
  factorLevels = NULL,
  sampleLevels = NULL,
  facetNames = NULL,
  plotgroup = FALSE,
  groupfun = mean,
  ...
)
```

Arguments

obj	phyloseq, phyloseq class or data.frame, (nrow sample * ncol feature (factor)) or the data.frame for geom_bar.

...	additional parameters, see ggplot
mapping	set of aesthetic mapping of ggplot2, default is NULL, if the data is the data.frame for geom_bar, the mapping should be set.
position	character, default is 'stack'.
stat	character, default is 'identity'.
width	numeric, the width of bar, default is 0.7.
topn	integer, the top number of abundance taxonomy(feature).
count	logical, whether show the relative abundance.
sampleda	data.frame, (nrow sample * ncol factor), the sample information, if the data doesn't contain the information.
factorLevels	vector or list, the levels of the factors (contained names e.g. list(group=c("B","A","C")) or c(group=c("B","A","C"))), adjust the order of facet, default is NULL, if you want to order the levels of factor, you can set this.
sampleLevels	vector, adjust the order of x axis e.g. c("sample2", "sample4", "sample3"), default is NULL.
facetNames	character, default is NULL.
plotgroup	logical, whether calculate the mean or median etc for each group, default is FALSE.
groupfun	character, how to calculate for feature in each group, the default is 'mean', this will plot the mean of feature in each group.

Value

barplot of tax

Author(s)

Shuangbin Xu

Examples

```
## Not run:
    library(ggplot2)
    data(test_otu_data)
    test_otu_data %<>% as.phyloseq()
    otubar <- ggbartax(test_otu_data) +
              xlab(NULL) + ylab("relative abundance(%)")

## End(Not run)
```

ggbox *A box or violin plot with significance test*

Description

A box or violin plot with significance test

Usage

```
ggbox(obj, factorNames, ...)

## S4 method for signature 'data.frame'
ggbox(
  obj,
  sampleda,
  factorNames,
  indexNames,
  geom = "boxplot",
  factorLevels = NULL,
  compare = TRUE,
  testmethod = "wilcox.test",
  signifmap = FALSE,
  p_textsize = 2,
  step_increase = 0.1,
  boxwidth = 0.2,
  facetnrow = 1,
  controlgroup = NULL,
  comparelist = NULL,
  ...
)

## S4 method for signature 'alphasample'
ggbox(obj, factorNames, ...)
```

Arguments

obj	object, alphasample or data.frame (row sample x column features).
factorNames	character, the names of factor contained in sampleda.
...	additional arguments, see also stat_signif.
sampleda	data.frame, sample information if obj is data.frame, the sampleda should be provided.
indexNames	character, the vector character, should be the names of features contained object.
geom	character, "boxplot" or "violin", default is "boxplot".
factorLevels	list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.

compare	logical, whether test the features among groups,default is TRUE.
testmethod	character, the method of test, default is 'wilcox.test'. see also stat_signif.
signifmap	logical, whether the pvalue are directly written a annotaion or asterisks are used instead, default is (pvalue) FALSE. see also stat_signif.
p_textsize	numeric, the size of text of pvalue or asterisks, default is 2.
step_increase	numeric, see also stat_signif, default is 0.1.
boxwidth	numeric, the width of boxplot when the geom is 'violin', default is 0.2.
facetnrow	integer, the nrow of facet, default is 1.
controlgroup	character, the names of control group, if it was set, the other groups will compare to it, default is NULL.
comparelist	list, the list of vector, default is NULL.

Value

a 'ggplot' plot object, a box or violine plot.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
library(magrittr)
otudafile <- system.file("extdata", "otu_tax_table.txt",
                         package="MicrobiotaProcess")
otuda <- read.table(otudafile, sep="\t",
                    header=TRUE, row.names=1,
                    check.names=FALSE, skip=1,
                    comment.char="")
samplefile <- system.file("extdata",
                          "sample_info.txt",
                          package="MicrobiotaProcess")
sampleda <- read.table(samplefile,
                       sep="\t", header=TRUE, row.names=1)
otuda <- otuda[sapply(otuda, is.numeric)] %>% t() %>%
         data.frame(check.names=FALSE)
set.seed(1024)
alphaobj1 <- get_alphaindex(otuda, sampleda=sampleda)
p1 <- ggbox(alphaobj1, factorNames="group")
data(test_otu_data)
test_otu_data %<>% as.phyloseq()
set.seed(1024)
alphaobj2 <- get_alphaindex(test_otu_data)
class(alphaobj2)
head(as.data.frame(alphaobj2))
p2 <- ggbox(alphaobj2, factorNames="group")
# set factor levels.
p3 <- ggbox(obj=alphaobj2, factorNames="group",
```

```
            factorLevels=list(group=c("M", "N", "B", "D")))
# set control group.
p4 <- ggbox(obj=alphaobj2, factorNames="group", controlgroup="B")
 set comparelist
p5 <- ggbox(obj=alphaobj2, factorNames="group",
            comparelist=list(c("B", "D"), c("B", "M"), c("B", "N")))

## End(Not run)
```

---

ggclust *plot the result of hierarchical cluster analysis for the samples*

Description

plot the result of hierarchical cluster analysis for the samples

Usage

```
ggclust(obj, ...)

## S3 method for class 'treedata'
ggclust(
  obj,
  layout = "rectangular",
  factorNames = NULL,
  factorLevels = NULL,
  pointsize = 2,
  fontsize = 2.6,
  hjust = -0.1,
  ...
)
```

Arguments

obj	R object, treedata object.
...	additional params, see also geom_tippoint
layout	character, the layout of tree, see also ggtree.
factorNames	character, default is NULL.
factorLevels	list, default is NULL.
pointsize	numeric, the size of point, default is 2.
fontsize	numeric, the size of text of tiplabel, default is 2.6.
hjust	numeric, default is -0.1

Value

the figures of hierarchical cluster.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
library(phyloseq)
library(ggtree)
library(ggplot2)
data(GlobalPatterns)
subGlobal <- subset_samples(GlobalPatterns,
         SampleType %in% c("Feces", "Mock", "Ocean", "Skin"))
hcsample <- get_clust(subGlobal, distmethod="jaccard",
                  method="hellinger", hclustmethod="average")
hc_p <- ggclust(hcsample, layout = "rectangular",
                pointsize=1, fontsize=0,
                factorNames=c("SampleType")) +
        theme_tree2(legend.position="right",
                    plot.title = element_text(face="bold", lineheight=25,hjust=0.5))

## End(Not run)
```

---

ggdiffbox *boxplot for the result of diff_analysis*

Description

boxplot for the result of diff_analysis

Usage

```
ggdiffbox(obj, ...)

## S4 method for signature 'diffAnalysisClass'
ggdiffbox(
  obj,
  geom = "boxplot",
  box_notch = TRUE,
  box_width = 0.05,
  dodge_width = 0.6,
  addLDA = TRUE,
  factorLevels = NULL,
  featurelist = NULL,
  removeUnknown = TRUE,
  colorlist = NULL,
  l_xlabtext = NULL,
  ...
)
```

Arguments

obj	object, diffAnalysisClass class.
...	additional arguments.
geom	character, "boxplot" or "violin", default is "boxplot".
box_notch	logical, see also ‘notch‘ of geom_boxplot, default is TRUE.
box_width	numeric, the width of boxplot, default is 0.05
dodge_width	numeric, the width of dodge of boxplot, default is 0.6.
addLDA	logical, whether add the plot to visulize the result of LDA, default is TRUE.
factorLevels	list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.
featurelist	vector, the character vector, the sub feature of originalD in diffAnalysisClass,default is NULL.
removeUnknown	logical, whether remove the unknown taxonomy, default is TRUE.
colorlist	character, the color vector, default is NULL.
l_xlabtext	character, the x axis text of left panel, default is NULL.

Value

a ’ggplot’ plot object, a box or violine plot for the result of diffAnalysisClass.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(kostic2012crc)
kostic2012crc %<>% as.phyloseq()
head(phyloseq::sample_data(kostic2012crc),3)
kostic2012crc <- phyloseq::rarefy_even_depth(kostic2012crc,
                 rngseed=1024)
table(phyloseq::sample_data(kostic2012crc)$DIAGNOSIS)
set.seed(1024)
diffres <- diff_analysis(kostic2012crc, classgroup="DIAGNOSIS",
                         mlfun="lda", filtermod="fdr",
                         firstcomfun = "kruskal.test",
                         firstalpha=0.05, strictmod=TRUE,
                         secondcomfun = "wilcox.test",
                         subclmin=3, subclwilc=TRUE,
                         secondalpha=0.01, ldascore=3)
library(ggplot2)
p <- ggdiffbox(diffres, box_notch=FALSE, l_xlabtext="relative abundance")
# set factor levels
p2 <- ggdiffbox(diffres, box_notch=FALSE, l_xlabtext="relative abundance",
                factorLevels=list(DIAGNOSIS=c("Tumor", "Healthy")))

## End(Not run)
```

ggdiffclade *plot the clade tree with highlight*

Description

plot results of different analysis or data.frame, contained hierarchical relationship or other classes,such like the tax_data of phyloseq.

Usage

```
ggdiffclade(obj, ...)

## S3 method for class 'data.frame'
ggdiffclade(
  obj,
  nodedf,
  factorName,
  size,
  layout = "radial",
  linewd = 0.6,
  bg.tree.color = "#bed0d1",
  bg.point.color = "#bed0d1",
  bg.point.stroke = 0.2,
  bg.point.fill = "white",
  skpointsize = 2,
  hilight.size = 0.2,
  alpha = 0.4,
  taxlevel = 5,
  cladetext = 2.5,
  tip.annot = TRUE,
  as.tiplab = TRUE,
  factorLevels = NULL,
  xlim = 12,
  removeUnknown = FALSE,
  reduce = FALSE,
  type = "species",
  ...
)

## S3 method for class 'diffAnalysisClass'
ggdiffclade(obj, size, removeUnknown = TRUE, ...)
```

Arguments

obj	object, diffAnalysisClass, the results of diff_analysis see also diff_analysis, or data.frame, contained hierarchical relationship or other classes.
...	additional parameters.

nodedf	data.frame, contained the tax and the factor information and(or pvalue).
factorName	character, the names of factor in nodedf.
size	the column name for mapping the size of points, default is 'pvalue'.
layout	character, the layout of ggtree, but only "rectangular", "roundrect", "ellipse", "radial", "slanted", "inward_circular" and "circular" in here, default is "radial".
linewd	numeric, the size of segment of ggtree, default is 0.6.
bg.tree.color	character, the line color of tree, default is '#bed0d1'.
bg.point.color	character, the color of margin of background node points of tree, default is '#bed0d1'.
bg.point.stroke	numeric, the margin thickness of point of background nodes of tree, default is 0.2 .
bg.point.fill	character, the point fill (since point shape is 21) of background nodes of tree, default is 'white'.
skpointsize	numeric, the point size of skeleton of tree, default is 2.
hilight.size	numeric, the margin thickness of high light clade, default is 0.2.
alpha	numeric, the alpha of clade, default is 0.4.
taxlevel	positive integer, the full text of clade, default is 5.
cladetext	numeric, the size of text of clade, default is 2.
tip.annot	logcial whether to replace the differential tip labels with shorthand, default is TRUE.
as.tiplab	logical, whether to display the differential tip labels with 'geom_tiplab' of 'ggtree', default is TRUE, if it is FALSE, it will use 'geom_text_repel' of 'ggrepel'.
factorLevels	list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.
xlim	numeric, the x limits, only works for 'inward_circular' layout, default is 12.
removeUnknown	logical, whether do not show unknown taxonomy, default is TRUE.
reduce	logical, whether remove the unassigned taxonomy, which will remove the clade of unassigned taxonomy, but the result of 'diff_analysis' should remove the unknown taxonomy, default is FALSE.
type	character, the type of datasets, default is "species", if the dataset is not about species, such as dataset of kegg function, you should set it to "others".

Value

figures of tax clade show the significant different feature.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(kostic2012crc)
kostic2012crc %<>% as.phyloseq()
head(phyloseq::sample_data(kostic2012crc),3)
kostic2012crc <- phyloseq::rarefy_even_depth(kostic2012crc,
                         rngseed=1024)
table(phyloseq::sample_data(kostic2012crc)$DIAGNOSIS)
set.seed(1024)
diffres <- diff_analysis(kostic2012crc, classgroup="DIAGNOSIS",
                         mlfun="lda", filtermod="fdr",
                         firstcomfun = "kruskal.test",
                         firstalpha=0.05, strictmod=TRUE,
                         secondcomfun = "wilcox.test",
                         subclmin=3, subclwilc=TRUE,
                         secondalpha=0.01, ldascore=3)
library(ggplot2)
diffcladeplot <- ggdiffclade(diffres,alpha=0.3, linewd=0.2,
                         skpointsize=0.4,
                         taxlevel=5) +
                 scale_fill_diff_cladogram(
                         values=c('#00AED7',
                                  '#FD9347'
                                  )
                 ) +
                 scale_size_continuous(range = c(1, 3))

## End(Not run)
```

---

ggdifftaxbar *significantly discriminative feature barplot*

Description

significantly discriminative feature barplot

Usage

```
ggdifftaxbar(obj, ...)

ggdiffbartaxa(obj, ...)

## S4 method for signature 'diffAnalysisClass'
ggdifftaxbar(
  obj,
  filepath = NULL,
  output = "biomarker_barplot",
  removeUnknown = TRUE,
  figwidth = 6,
```

```
  figheight = 3,
  ylabel = "relative abundance",
  format = "pdf",
  dpi = 300,
  ...
)

## S3 method for class 'featureMeanMedian'
ggdifftaxbar(
  obj,
  featurename,
  classgroup,
  subclass,
  xtextsize = 3,
  factorLevels = NULL,
  coloslist = NULL,
  ylabel = "relative abundance",
  ...
)
```

Arguments

obj	object, diffAnalysisClass see also diff_analysis or feMeanMedian class, see also get_mean_median.
...	additional arguments.
filepath	character, default is NULL, meaning current path.
output	character, the output dir name, default is "biomarker_barplot".
removeUnknown	logical, whether do not show unknown taxonomy, default is TRUE.
figwidth	numeric, the width of figures, default is 6.
figheight	numeric, the height of figures, default is 3.
ylabel	character, the label of y, default is 'relative abundance'.
format	character, the format of figure, default is pdf, png, tiff also be supported.
dpi	numeric, the dpi of output, default is 300.
featurename	character, the feature name, contained at the objet.
classgroup	character, factor name.
subclass	character, factor name.
xtextsize	numeric, the size of axis x label, default is 3.
factorLevels	list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.
coloslist	vector, color vector, if the input is phyloseq, you should use this to adjust the color, not scale_color_manual.

Value

the figures of features show the distributions in samples.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(kostic2012crc)
kostic2012crc %<>% as.phyloseq()
head(phyloseq::sample_data(kostic2012crc),3)
kostic2012crc <- phyloseq::rarefy_even_depth(kostic2012crc,
                                rngseed=1024)
table(phyloseq::sample_data(kostic2012crc)$DIAGNOSIS)
set.seed(1024)
diffres <- diff_analysis(kostic2012crc, classgroup="DIAGNOSIS",
                         mlfun="lda", filtermod="fdr",
                         firstcomfun = "kruskal.test",
                         firstalpha=0.05, strictmod=TRUE,
                         secondcomfun = "wilcox.test",
                         subclmin=3, subclwilc=TRUE,
                         secondalpha=0.01, ldascore=3)
ggdifftaxbar(diffres, output="biomarker_barplot")

## End(Not run)
```

---

ggeffectsize *visualization of effect size by the Linear Discriminant Analysis or randomForest*

Description

visualization of effect size by the Linear Discriminant Analysis or randomForest

Usage

```
ggeffectsize(obj, ...)

## S3 method for class 'data.frame'
ggeffectsize(
  obj,
  factorName,
  effectsizename,
  factorLevels = NULL,
  linecolor = "grey50",
  linewidth = 0.4,
  lineheight = 0.2,
  pointsize = 1.5,
  setFacet = TRUE,
  ...
```

```
)

## S3 method for class 'diffAnalysisClass'
ggeffectsize(obj, removeUnknown = TRUE, setFacet = TRUE, ...)
```

Arguments

obj	object, diffAnalysisClass see diff_analysis, or data.frame, contained effect size and the group information.
...	additional arguments.
factorName	character, the column name contained group information in data.frame.
effectsizename	character, the column name contained effect size information.
factorLevels	list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.
linecolor	character, the color of horizontal error bars, default is grey50.
linewidth	numeric, the width of horizontal error bars, default is 0.4.
lineheight	numeric, the height of horizontal error bars, default is 0.2.
pointsize	numeric, the size of points, default is 1.5.
setFacet	logical, whether use facet to plot, default is TRUE.
removeUnknown	logical, whether do not show unknown taxonomy, default is TRUE.

Value

the figures of effect size show the LDA or MDA (MeanDecreaseAccuracy).

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(kostic2012crc)
kostic2012crc %<>% as.phyloseq()
head(phyloseq::sample_data(kostic2012crc),3)
kostic2012crc <- phyloseq::rarefy_even_depth(kostic2012crc,rngseed=1024)
table(phyloseq::sample_data(kostic2012crc)$DIAGNOSIS)
set.seed(1024)
diffres <- diff_analysis(kostic2012crc, classgroup="DIAGNOSIS",
                         mlfun="lda", filtermod="fdr",
                         firstcomfun = "kruskal.test",
                         firstalpha=0.05, strictmod=TRUE,
                         secondcomfun = "wilcox.test",
                         subclmin=3, subclwilc=TRUE,
                         secondalpha=0.01, ldascore=3)
library(ggplot2)
effectplot <- ggeffectsize(diffres) +
              scale_color_manual(values=c('#00AED7',
```

```
                                    '#FD9347',
                                    '#C1E168'))+
                theme_bw()+
                theme(strip.background=element_rect(fill=NA),
                      panel.spacing = unit(0.2, "mm"),
                      panel.grid=element_blank(),
                      strip.text.y=element_blank())

## End(Not run)
```

---

ggordpoint *ordination plotter based on ggplot2.*

Description

ordination plotter based on ggplot2.

Usage

```
ggordpoint(obj, ...)

## Default S3 method:
ggordpoint(
  obj,
  pc = c(1, 2),
  mapping = NULL,
  sampleda = NULL,
  factorNames = NULL,
  factorLevels = NULL,
  poinsize = 2,
  linesize = 0.3,
  arrowsize = 1.5,
  arrowlinecolour = "grey",
  ellipse = FALSE,
  showsample = FALSE,
  ellipse_pro = 0.9,
  ellipse_alpha = 0.2,
  ellipse_linewd = 0.5,
  ellipse_lty = 3,
  biplot = FALSE,
  topn = 5,
  settheme = TRUE,
  speciesannot = FALSE,
  fontsize = 2.5,
  labelfactor = NULL,
  stroke = 0.1,
  fontface = "bold.italic",
  fontfamily = "sans",
```

```
  textlinesize = 0.02,
  ...
)

## S3 method for class 'pcasample'
ggordpoint(obj, ...)
```

Arguments

obj	prcomp class or pcasample class,
...	additional parameters, see geom_text_repel.
pc	integer vector, the component index.
mapping	set of aesthetic mapping of ggplot2, default is NULL when your want to set it by yourself, only alpha can be setted, and the first element of factorNames has been setted to map 'fill', and the second element of factorNames has been setted to map 'starshape', you can add 'scale_starshape_manual' of 'ggstar' to set the shapes.
sampleda	data.frame, nrow sample * ncol factors, default is NULL.
factorNames	vector, the names of factors contained sampleda.
factorLevels	list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.
poinsize	numeric, the size of point, default is 2.
linesize	numeric, the line size of segment, default is 0.3.
arrowsize	numeric, the size of arrow, default is 1.5.
arrowlinecolour	character, the color of segment, default is grey.
ellipse	logical, whether add confidence ellipse to ordinary plot, default is FALSE.
showsample	logical, whether show the labels of sample, default is FALSE.
ellipse_pro	numeric, confidence value for the ellipse, default is 0.9.
ellipse_alpha	numeric, the alpha of ellipse, default is 0.2.
ellipse_linewd	numeric, the width of ellipse line, default is 0.5.
ellipse_lty	integer, the type of ellipse line, default is 3
biplot	logical, whether plot the species, default is FALSE.
topn	integer or vector, the number species have top important contribution, default is 5.
settheme	logical, whether set the theme for the plot, default is TRUE.
speciesannot	logical, whether plot the species, default is FALSE.
fontsize	numeric, the size of text, default is 2.5.
labelfactor	character, the factor want to be show in label, default is NULL.
stroke	numeric, the line size of points, default is 0.1.
fontface	character, the font face, default is "blod.italic".
fontfamily	character, the font family, default is "sans".
textlinesize	numeric, the segment size in geom_text_repel.

Value

point figures of PCA or PCoA.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
library(phyloseq)
data(GlobalPatterns)
subGlobal <- subset_samples(GlobalPatterns,
         SampleType %in% c("Feces", "Mock", "Ocean", "Skin"))
pcares <- get_pca(subGlobal, method="hellinger")
pcaplot <- ggordpoint(pcares, biplot=TRUE,
                    speciesannot=TRUE,
                     factorNames=c("SampleType"), ellipse=TRUE)

## End(Not run)
```

---

ggrarecurve *Rarefaction alpha index*

Description

Rarefaction alpha index

Usage

```
ggrarecurve(obj, ...)

## S3 method for class 'phyloseq'
ggrarecurve(obj, chunks = 400, factorLevels = NULL, ...)

## S3 method for class 'data.frame'
ggrarecurve(obj, sampleda, factorLevels, chunks = 400, ...)

## S3 method for class 'rarecurve'
ggrarecurve(
  obj,
  indexNames = "Observe",
  linesize = 0.5,
  facetnrow = 1,
  shadow = TRUE,
  factorNames,
  se = FALSE,
  method = "lm",
```

```
  formula = y ~ log(x),
  ...
)
```

Arguments

obj	phyloseq, phyloseq class or data.frame shape of data.frame (nrow sample * ncol feature (+ factor)).
...	additional parameters, see also ggplot2{ggplot}.
chunks	integer, the number of subsample in a sample, default is 400.
factorLevels	list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.
sampleda	data.frame, (nrow sample * ncol factor)
indexNames	character, default is "Observe", only for "Observe", "Chao1", "ACE".
linesize	integer, default is 0.5.
facetnrow	integer, the nrow of facet, default is 1.
shadow	logical, whether merge samples with group (factorNames) and display the ribbon of group, default is TRUE.
factorNames	character, default is missing.
se	logical, default is FALSE.
method	character, default is lm.
formula	formula, default is 'y ~ log(x)'

Value

figure of rarefaction curves

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(test_otu_data)
test_otu_data %<>% as.phyloseq()
library(ggplot2)
prare <- ggrarecurve(test_otu_data,
                indexNames=c("Observe","Chao1","ACE"),
                shadow=FALSE,
                factorNames="group"
        ) +
        theme(legend.spacing.y=unit(0.02,"cm"),
                legend.text=element_text(size=6))

## End(Not run)
```

ImportDada2 *Import function to load the feature table and taxonomy table of dada2*

Description

the function can import the ouput of dada2, and generated the phyloseq obj contained the argument class.

Usage

```
import_dada2(seqtab, taxatab = NULL, reftree = NULL, sampleda = NULL, ...)

mp_import_dada2(seqtab, taxatab = NULL, reftree = NULL, sampleda = NULL, ...)
```

Arguments

seqtab	matrix, feature table, the output of removeBimeraDenovo.
taxatab	matrix, a taxonomic table, the output of assignTaxonomy, or the ouput of addSpecies.
reftree	phylo, treedata or character, the treedata or phylo class of tree, or the tree file.
sampleda	data.frame or character, the data.frame of sample information, or the file of sample information, nrow samples X ncol factors.
...	additional parameters.

Value

phyloseq class contained the argument class.

Author(s)

Shuangbin Xu

Examples

```
seqtabfile <- system.file("extdata", "seqtab.nochim.rds",
                          package="MicrobiotaProcess")
taxafile <- system.file("extdata", "taxa_tab.rds",
                        package="MicrobiotaProcess")
seqtab <- readRDS(seqtabfile)
taxa <- readRDS(taxafile)
sampleda <- system.file("extdata", "mouse.time.dada2.txt",
                        package="MicrobiotaProcess")
mpse <- mp_import_dada2(seqtab=seqtab, taxatab=taxa,
                    sampleda=sampleda)
mpse
```

ImportQiime2 *Import function to load the output of qiime2.*

Description

The function was designed to import the output of qiime2 and convert them to phyloseq class.

Usage

```
import_qiime2(
  otuqza,
  taxaqza = NULL,
  mapfilename = NULL,
  refseqqza = NULL,
  treeqza = NULL,
  parallel = FALSE,
  ...
)

mp_import_qiime2(
  otuqza,
  taxaqza = NULL,
  mapfilename = NULL,
  refseqqza = NULL,
  treeqza = NULL,
  parallel = FALSE,
  ...
)
```

Arguments

otuqza	character, the file contained otu table, the ouput of qiime2.
taxaqza	character, the file contained taxonomy, the ouput of qiime2, default is NULL.
mapfilename	character, the file contained sample information, the tsv format, default is NULL.
refseqqza	character, the file contained reference sequences or the XStringSet object, default is NULL.
treeqza	character, the file contained the tree file or treedata object, which is the result parsed by functions of treeio, default is NULL.
parallel	logical, whether parsing the column of taxonomy multi-parallel, default is FALSE.
...	additional parameters.

Value

MPSE-class or phyloseq-class contained the argument class.

Author(s)

Shuangbin Xu

Examples

```
otuqzafile <- system.file("extdata", "table.qza",
                          package="MicrobiotaProcess")
taxaqzafile <- system.file("extdata", "taxa.qza",
                           package="MicrobiotaProcess")
mapfile <- system.file("extdata", "metadata_qza.txt",
                       package="MicrobiotaProcess")
mpse <- mp_import_qiime2(otuqza=otuqzafile, taxaqza=taxaqzafile,
                         mapfilename=mapfile)
mpse
```

---

mouse.time.mpse *(Data) An example data*

Description

This is a MPSE object example data.

MPSE *Construct a MPSE object*

Description

Construct a MPSE object

Usage

```
MPSE(
  assays,
  colData = NULL,
  otutree = NULL,
  taxatree = NULL,
  refseq = NULL,
  ...
)
```

Arguments

assays	A 'list' or 'SimpleList' of matrix-like elements All elements of the list must have the same dimensions, we also recommend they have names, e.g. list(Abundance=xx1, RareAbundance=xx2).
colData	An optional DataFrame describing the samples.
otutree	A treedata object of tidytree package, the result parsed by the functions of treeio.
taxatree	A treedata object of tidytree package, the result parsed by the functions of treeio.
refseq	A XStingSet object of Biostrings package, the result parsed by the readDNAStringSet or readAAStringSet of Biostrings.
...	additional parameters, see also the usage of SummarizedExperiment.

Value

MPSE object

Examples

```
set.seed(123)
xx <- matrix(abs(round(rnorm(100, sd=4), 0)), 10)
xx <- data.frame(xx)
rownames(xx) <- paste0("row", seq_len(10))
mpse <- MPSE(assays=xx)
mpse
```

MPSE-accessors *MPSE accessors*

Description

MPSE accessors

Usage

```
## S4 method for signature 'MPSE,ANY,ANY,ANY'
x[i, j, ..., drop = TRUE]

## S4 replacement method for signature 'MPSE,DataFrame'
colData(x, ...) <- value

## S4 replacement method for signature 'MPSE,NULL'
colData(x, ...) <- value

tax_table(object)

## S4 method for signature 'MPSE'
tax_table(object)
```

```
## S4 method for signature 'tbl_mpse'
tax_table(object)

## S4 method for signature 'grouped_df_mpse'
tax_table(object)

otutree(x, ...)

## S4 method for signature 'MPSE'
otutree(x, ...)

## S4 method for signature 'tbl_mpse'
otutree(x, ...)

## S4 method for signature 'MPSE'
otutree(x, ...)

otutree(x, ...) <- value

## S4 replacement method for signature 'MPSE,treedata'
otutree(x, ...) <- value

## S4 replacement method for signature 'MPSE,phylo'
otutree(x, ...) <- value

## S4 replacement method for signature 'MPSE,NULL'
otutree(x, ...) <- value

## S4 replacement method for signature 'tbl_mpse,treedata'
otutree(x, ...) <- value

## S4 replacement method for signature 'grouped_df_mpse,treedata'
otutree(x, ...) <- value

## S4 replacement method for signature 'tbl_mpse,NULL'
otutree(x, ...) <- value

## S4 replacement method for signature 'grouped_df_mpse,NULL'
otutree(x, ...) <- value

taxatree(x, ...)

## S4 method for signature 'MPSE'
taxatree(x, ...)

## S4 method for signature 'tbl_mpse'
taxatree(x, ...)
```

```
## S4 method for signature 'grouped_df_mpse'
taxatree(x, ...)

taxatree(x, ...) <- value

## S4 replacement method for signature 'MPSE,treedata'
taxatree(x, ...) <- value

## S4 replacement method for signature 'MPSE,NULL'
taxatree(x, ...) <- value

## S4 replacement method for signature 'tbl_mpse,treedata'
taxatree(x, ...) <- value

## S4 replacement method for signature 'tbl_mpse,NULL'
taxatree(x, ...) <- value

## S4 replacement method for signature 'grouped_df_mpse,treedata'
taxatree(x, ...) <- value

## S4 replacement method for signature 'grouped_df_mpse,NULL'
taxatree(x, ...) <- value

taxonomy(x, ...) <- value

## S4 replacement method for signature 'MPSE,data.frame'
taxonomy(x, ...) <- value

## S4 replacement method for signature 'MPSE,matrix'
taxonomy(x, ...) <- value

## S4 replacement method for signature 'MPSE,taxonomyTable'
taxonomy(x, ...) <- value

## S4 replacement method for signature 'MPSE,NULL'
taxonomy(x, ...) <- value

refsequence(x, ...)

## S4 method for signature 'MPSE'
refsequence(x, ...)

refsequence(x, ...) <- value

## S4 replacement method for signature 'MPSE,XStringSet'
refsequence(x, ...) <- value
```

```
## S4 replacement method for signature 'MPSE,NULL'
refsequence(x, ...) <- value

## S4 replacement method for signature 'MPSE'
rownames(x) <- value
```

Arguments

x	MPSE object
i, j, ...	Indices specifying elements to extract or replace. Indices are 'numeric' or 'character' vectors or empty (missing) or NULL. Numeric values are coerced to integer as by 'as.integer' (and hence truncated towards zero). Character vectors will be matched to the 'names' of the object (or for matrices/arrays, the 'dimnames')
drop	logical If 'TRUE' the result is coerced to the lowest possible dimension (see the examples). This only works for extracting elements, not for the replacement.
value	XStringSet object or NULL
object	parameter of tax_table, R object, MPSE class in here.

Value

taxonomyTable class

`MPSE-class` *MPSE class*

Description

MPSE class

Slots

`otutree` A treedata object of tidytree package or NULL.

`taxatree` A treedata object of tidytree package or NULL.

`refseq` A XStringSet object of Biostrings package or NULL.

`...` Other slots from `SummarizedExperiment`

`mp_adonis` *Permutational Multivariate Analysis of Variance Using Distance Matrices for MPSE or tbl_mpse object*

Description

Permutational Multivariate Analysis of Variance Using Distance Matrices for MPSE or tbl_mpse object

Usage

```
mp_adonis(
  .data,
  .abundance,
  .formula,
  distmethod = "bray",
  action = "get",
  permutations = 999,
  seed = 123,
  ...
)

## S4 method for signature 'MPSE'
mp_adonis(
  .data,
  .abundance,
  .formula,
  distmethod = "bray",
  action = "get",
  permutations = 999,
  seed = 123,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_adonis(
  .data,
  .abundance,
  .formula,
  distmethod = "bray",
  action = "get",
  permutations = 999,
  seed = 123,
  ...
)

## S4 method for signature 'grouped_df_mpse'
```

```
mp_adonis(
  .data,
  .abundance,
  .formula,
  distmethod = "bray",
  action = "get",
  permutations = 999,
  seed = 123,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of abundance to be calculated.
.formula	Model formula right hand side gives the continuous variables or factors, and keep left empty, such as ~ group, it is required.
distmethod	character the method to calculate pairwise distances, default is 'bray'.
action	character "add" joins the cca result to the object, "only" return a non-redundant tibble with the cca result. "get" return 'cca' object can be analyzed using the related vegan funtion.
permutations	the number of permutations required, default is 999.
seed	a random seed to make the adonis analysis reproducible, default is 123.
...	additional parameters see also 'adonis2' of vegan.

Value

update object according action argument

Author(s)

Shuangbin Xu

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %>%
  mp_decostand(
     .abundance=Abundance,
     method="hellinger") %>%
  mp_adonis(.abundance=hellinger,
            .formula=~time,
            distmethod="bray",
            permutations=999, # for more robust, set it to 9999.
            action="get")
```

mp_aggregate *aggregate the assays with the specific group of sample and fun.*

Description

aggregate the assays with the specific group of sample and fun.

Usage

```
mp_aggregate(.data, .abundance, .group, fun = sum, keep_colData = TRUE, ...)

## S4 method for signature 'MPSE'
mp_aggregate(.data, .abundance, .group, fun = sum, keep_colData = TRUE, ...)
```

Arguments

.data	MPSE object, required
.abundance	the column names of abundance, default is Abundance.
.group	the column names of sample meta-data, required
fun	a function to compute the summary statistics, default is sum.
keep_colData	logical whether to keep the sample meta-data with .group as row names, default is TRUE.
...	additional parameters, see also aggregate.

Value

a new object with .group as column names in assays

Examples

```
## Not run:
data(mouse.time.mpse)
newmpse <- mouse.time.mpse %>%
           mp_aggregate(.group = time)
newmpse

## End(Not run)
```

mp_aggregate_clade *calculate the mean/median (relative) abundance of internal nodes according to their children tips.*

Description

calculate the mean/median (relative) abundance of internal nodes according to their children tips.

Usage

```
mp_aggregate_clade(
  .data,
  .abundance = NULL,
  force = FALSE,
  relative = TRUE,
  aggregate_fun = c("mean", "median", "geometric.mean"),
  action = "get",
  ...
)

## S4 method for signature 'MPSE'
mp_aggregate_clade(
  .data,
  .abundance = NULL,
  force = FALSE,
  relative = TRUE,
  aggregate_fun = c("mean", "median", "geometric.mean"),
  action = "get",
  ...
)

## S4 method for signature 'tbl_mpse'
mp_aggregate_clade(
  .data,
  .abundance = NULL,
  force = FALSE,
  relative = TRUE,
  aggregate_fun = c("mean", "median", "geometric.mean"),
  action = "get",
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_aggregate_clade(
  .data,
  .abundance = NULL,
  force = FALSE,
```

```
  relative = TRUE,
  aggregate_fun = c("mean", "median", "geometric.mean"),
  action = "get",
  ...
)
```

Arguments

.data	MPSE object which must contain otutree slot, required
.abundance	the column names of abundance.
force	logical whether calculate the (relative) abundance forcibly when the abundance is not be rarefied, default is FALSE.
relative	logical whether calculate the relative abundance.
aggregate_fun	function the method to calculate the (relative) abundance of internal nodes according to their children tips, default is 'mean', other options are 'median', 'geometric.mean'.
action	character, "add" joins the new information to the otutree slot if it exists (default). In addition, "only" return a non-redundant tibble with the just new information. "get" return a new 'mpse', which the features is the internal nodes.
...	additional parameters, meaningless now.

Value

a object according to 'action' argument.

Examples

```
## Not run:
  suppressPackageStartupMessages(library(curatedMetagenomicData))
  xx <- curatedMetagenomicData('ZellerG_2014.relative_abundance', dryrun=F)
  xx[[1]] %>% as.mpse -> mpse
  otu.tree <- mpse %>%
    mp_aggregate_clade(
      .abundance = Abundance,
      force = TRUE,
      relative = FALSE,
      action = 'get' # other option is 'add' or 'only'.
    )
  otu.tree

## End(Not run)
```

mp_anosim *Analysis of Similarities (ANOSIM) with MPSE or tbl_mpse object*

Description

Analysis of Similarities (ANOSIM) with MPSE or tbl_mpse object

Usage

```
mp_anosim(
  .data,
  .abundance,
  .group,
  distmethod = "bray",
  action = "add",
  permutations = 999,
  seed = 123,
  ...
)

## S4 method for signature 'MPSE'
mp_anosim(
  .data,
  .abundance,
  .group,
  distmethod = "bray",
  action = "add",
  permutations = 999,
  seed = 123,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_anosim(
  .data,
  .abundance,
  .group,
  distmethod = "bray",
  action = "add",
  permutations = 999,
  seed = 123,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_anosim(
  .data,
```

```
  .abundance,
  .group,
  distmethod = "bray",
  action = "add",
  permutations = 999,
  seed = 123,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of abundance to be calculated.
.group	The name of the column of the sample group information.
distmethod	character the method to calculate pairwise distances, default is 'bray'.
action	character "add" joins the ANOSIM result to internal attribute of the object, "only" and "get" return 'anosim' object can be analyzed using the related vegan funtion.
permutations	the number of permutations required, default is 999.
seed	a random seed to make the ANOSIM analysis reproducible, default is 123.
...	additional parameters see also 'anosim' of vegan.

Value

update object according action argument

Author(s)

Shuangbin Xu

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %<>%
  mp_decostand(.abundance=Abundance)
# action = "get" will return a anosim object
mouse.time.mpse %>%
  mp_anosim(.abundance=hellinger, .group=time, action="get")
# action = "only" will return a tbl_df that can be as the input of ggplot2.
library(ggplot2)
tbl <- mouse.time.mpse %>%
       mp_anosim(.abundance=hellinger,
                 .group=time,
                 permutations=999, # for more robust, set it to 9999
                 action="only")
tbl
tbl %>%
ggplot(aes(x=class, y=rank, fill=class)) +
geom_boxplot(notch=TRUE, varwidth = TRUE)
```

mp_balance_clade *Calculating the balance score of internal nodes (clade) according to the geometric.mean/mean/median abundance of their binary children tips.*

Description

Calculating the balance score of internal nodes (clade) according to the geometric.mean/mean/median abundance of their binary children tips.

Usage

```
mp_balance_clade(
  .data,
  .abundance = NULL,
  force = FALSE,
  relative = TRUE,
  balance_fun = c("geometric.mean", "mean", "median"),
  pseudonum = 0.001,
  action = "get",
  ...
)

## S4 method for signature 'MPSE'
mp_balance_clade(
  .data,
  .abundance = NULL,
  force = FALSE,
  relative = TRUE,
  balance_fun = c("geometric.mean", "mean", "median"),
  pseudonum = 0.001,
  action = "get",
  ...
)

## S4 method for signature 'tbl_mpse'
mp_balance_clade(
  .data,
  .abundance = NULL,
  force = FALSE,
  relative = TRUE,
  balance_fun = c("geometric.mean", "mean", "median"),
  pseudonum = 0.001,
  action = "get",
  ...
)
```

```
## S4 method for signature 'grouped_df_mpse'
mp_balance_clade(
  .data,
  .abundance = NULL,
  force = FALSE,
  relative = TRUE,
  balance_fun = c("geometric.mean", "mean", "median"),
  pseudonum = 0.001,
  action = "get",
  ...
)
```

Arguments

.data	MPSE object which must contain otutree slot, required
.abundance	the column names of abundance.
force	logical whether calculate the (relative) abundance forcibly when the abundance is not be rarefied, default is FALSE.
relative	logical whether calculate the relative abundance.
balance_fun	function the method to calculate the (relative) abundance of internal nodes according to their children tips, default is 'geometric.mean', other options are 'mean' and 'median'.
pseudonum	numeric add a pseudo numeric to avoid the error of division in calculation, default is 0.001 .
action	character, "add" joins the new information to the otutree slot if it exists (default). In addition, "only" return a non-redundant tibble with the just new information. "get" return a new 'MPSE' object, and the 'OTU' column is the internal nodes and 'Abundance' column is the balance scores.
...	additional parameters, meaningless now.

Value

a object according to 'action' argument.

References

Morton JT, Sanders J, Quinn RA, McDonald D, Gonzalez A, Vázquez-Baeza Y, Navas-Molina JA, Song SJ, Metcalf JL, Hyde ER, Lladser M, Dorrestein PC, Knight R. 2017. Balance trees reveal microbial niche differentiation. mSystems 2:e00162-16. https://doi.org/10.1128/mSystems.00162-16.

Justin D Silverman, Alex D Washburne, Sayan Mukherjee, Lawrence A David. A phylogenetic transform enhances analysis of compositional microbiota data. eLife 2017;6:e21887. https://doi.org/10.7554/eLife.21887.001

Examples

```
## Not run:
  suppressPackageStartupMessages(library(curatedMetagenomicData))
```

```
xx <- curatedMetagenomicData('ZellerG_2014.relative_abundance', dryrun=F)
xx[[1]] %>% as.mpse -> mpse
mpse.balance.clade <- mpse %>%
  mp_balance_clade(
    .abundance = Abundance,
    force = TRUE,
    relative = FALSE,
    action = 'get',
    pseudonum = .01
  )
mpse.balance.clade

# Performing the Euclidean distance or PCA.

mpse.balance.clade %>%
  mp_cal_dist(.abundance = Abundance, distmethod = 'euclidean') %>%
  mp_plot_dist(.distmethod = 'euclidean', .group = disease, group.test = T)

mpse.balance.clade %>%
 mp_adonis(.abundance = Abundance, .formula=~disease, distmethod = 'euclidean', permutation = 9999)

mpse.balance.clade %>%
  mp_cal_pca(.abundance = Abundance) %>%
  mp_plot_ord(.group = disease)

# Detecting the signal balance nodes.
mpse.balance.clade %>% mp_diff_analysis(
    .abundance = Abundance,
    force = TRUE,
    relative = FALSE,
    .group = disease,
    fc.method = 'compare_mean'
)

## End(Not run)
```

`mp_cal_abundance` *Calculate the (relative) abundance of each taxonomy class for each sample or group.*

Description

Calculate the (relative) abundance of each taxonomy class for each sample or group.

Usage

```
mp_cal_abundance(
  .data,
  .abundance = NULL,
  .group = NULL,
```

```
  relative = TRUE,
  action = "add",
  force = FALSE,
  ...
)

## S4 method for signature 'MPSE'
mp_cal_abundance(
  .data,
  .abundance = NULL,
  .group = NULL,
  relative = TRUE,
  action = "add",
  force = FALSE,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_cal_abundance(
  .data,
  .abundance = NULL,
  .group = NULL,
  relative = TRUE,
  action = "add",
  force = FALSE,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_cal_abundance(
  .data,
  .abundance = NULL,
  .group = NULL,
  relative = TRUE,
  action = "add",
  force = FALSE,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of otu abundance to be calculated
.group	the name of group to be calculated.
relative	logical whether calculate the relative abundance.
action	character, "add" joins the new information to the taxatree and otutree if they exists (default). In addition, All taxonomy class will be added the taxatree,

	and OTU (tip) information will be added to the otutree."only" return a non-redundant tibble with the just new information. "get" return 'taxatree' slot which is a treedata object.
force	logical whether calculate the relative abundance forcibly when the abundance is not be rarefied, default is FALSE.
...	additional parameters.

Value

update object or tibble according the 'action'

Author(s)

Shuangbin Xu

See Also

[mp_plot_abundance()] and [mp_extract_abundance()]

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %<>%
  mp_rrarefy()
mouse.time.mpse
mouse.time.mpse %<>%
  mp_cal_abundance(.abundance=RareAbundance, action="add") %>%
  mp_cal_abundance(.abundance=RareAbundance, .group=time, action="add")
mouse.time.mpse
library(ggplot2)
f <- mouse.time.mpse %>%
     mp_plot_abundance(
        .abundance=RelRareAbundanceBySample,
        .group = time,
        taxa.class = "Phylum",
        topn = 20,
        geom = "heatmap",
        feature.dist = "bray",
        feature.hclust = "average"
     ) %>%
     set_scale_theme(
        x = scale_fill_manual(values=c("orange", "deepskyblue")),
        aes_var = time
     )
f
p1 <- mouse.time.mpse %>%
      mp_plot_abundance(.abundance=RelRareAbundanceBySample,
                        .group=time, taxa.class="Phylum",
                        topn=20, order.by.feature = "p__Firmicutes",
                        width = 4/5
                        )
```

```
p2 <- mouse.time.mpse %>%
      mp_plot_abundance(.abundance = RareAbundance,
                        .group = time,
                        taxa.class = Phylum,
                        topn = 20,
                        relative = FALSE,
                        force = TRUE,
                        order.by.feature = TRUE
                        )
p1 / p2
# Or you can also extract the result and visulize it with ggplot2 and ggplot2-extension
## Not run:
tbl <- mouse.time.mpse %>%
       mp_extract_abundance(taxa.class="Class", topn=10)
tbl
library(ggplot2)
library(ggalluvial)
library(dplyr)
tbl %<>%
  tidyr::unnest(cols=RareAbundanceBySample)
tbl
p <- ggplot(data=tbl,
            mapping=aes(x=Sample,
                        y=RelRareAbundanceBySample,
                        alluvium=label,
                        fill=label)
     ) +
     geom_flow(stat="alluvium", lode.guidance = "frontback", color = "darkgray") +
     geom_stratum(stat="alluvium") +
     labs(x=NULL, y="Relative Abundance (%)") +
     scale_fill_brewer(name="Class", type = "qual", palette = "Paired") +
     facet_grid(cols=vars(time), scales="free_x", space="free") +
     theme(axis.text.x=element_text(angle=-45, hjust=0))
p

## End(Not run)
```

`mp_cal_alpha` *calculate the alpha index with MPSE or tbl_mpse*

Description

calculate the alpha index with MPSE or tbl_mpse

Usage

```
mp_cal_alpha(
  .data,
  .abundance = NULL,
  action = c("add", "only", "get"),
```

```
  force = FALSE,
  ...
)

## S4 method for signature 'MPSE'
mp_cal_alpha(.data, .abundance = NULL, action = "add", force = FALSE, ...)

## S4 method for signature 'tbl_mpse'
mp_cal_alpha(.data, .abundance = NULL, action = "add", force = FALSE, ...)

## S4 method for signature 'grouped_df_mpse'
mp_cal_alpha(.data, .abundance = NULL, action = "add", force = FALSE, ...)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	The column name of OTU abundance column to be calculate
action	character it has three options, "add" joins the new information to the input tbl (default), "only" return a non-redundant tibble with the just new information, ang 'get' return a 'alphasample' object.
force	logical whether calculate the alpha index even the '.abundance' is not rarefied, default is FALSE.
...	additional arguments

Value

update object or other (refer to action)

Author(s)

Shuangbin Xu

See Also

[mp_plot_alpha()]

Examples

```
data(mouse.time.mpse)
mpse <- mouse.time.mpse %>%
        mp_rrarefy() %>%
        mp_cal_alpha(.abundance=RareAbundance)
mpse
p <- mpse %>% mp_plot_alpha(.group=time, .alpha=c(Observe, Shannon, Pielou))
p
# Or you can extract the result and visualize it with ggplot2 and ggplot2-extensions
## Not run:
tbl <- mpse %>%
       mp_extract_sample()
```

```
tbl
tbl %<>%
  tidyr::pivot_longer(cols=!c("Sample", "time"), names_to="measure", values_to="alpha")
tbl
library(ggplot2)
library(ggsignif)
p <- ggplot(data=tbl, aes(x=time, y=alpha, fill=time)) +
     geom_violin(color=NA, trim=FALSE) +
     geom_boxplot(aes(color=time), fill=NA, width=0.2) +
     geom_jitter(shape=21, width = .1) +
     geom_signif(comparisons=list(c("Early", "Late")), test="wilcox.test", textsize=2) +
     facet_wrap(facet=vars(measure), scales="free_y", nrow=1) +
     scale_fill_manual(values=c("#00A087FF", "#3C5488FF")) +
     scale_color_manual(values=c("#00A087FF", "#3C5488FF"))
p

## End(Not run)
```

`mp_cal_cca` *[Partial] [Constrained] Correspondence Analysis with MPSE or tbl_mpse object*

Description

[Partial] [Constrained] Correspondence Analysis with MPSE or tbl_mpse object

Usage

```
mp_cal_cca(.data, .abundance, .formula = NULL, .dim = 3, action = "add", ...)

## S4 method for signature 'MPSE'
mp_cal_cca(.data, .abundance, .formula = NULL, .dim = 3, action = "add", ...)

## S4 method for signature 'tbl_mpse'
mp_cal_cca(.data, .abundance, .formula = NULL, .dim = 3, action = "add", ...)

## S4 method for signature 'grouped_df_mpse'
mp_cal_cca(.data, .abundance, .formula = NULL, .dim = 3, action = "add", ...)
```

Arguments

`.data`	MPSE or tbl_mpse object
`.abundance`	the name of abundance to be calculated.
`.formula`	Model formula right hand side gives the constraining variables, and conditioning variables can be given within a special function 'Condition' and keep left empty, such as ~ A + B or ~ A + Condition(B), default is NULL.
`.dim`	integer The number of dimensions to be returned, default is 3.

action	character "add" joins the cca result to the object, "only" return a non-redundant tibble with the cca result. "get" return 'cca' object can be analyzed using the related vegan funtion.
...	additional parameters see also 'cca' of vegan.

Value

update object according action argument

Author(s)

Shuangbin Xu

Examples

```
library(vegan)
data(varespec, varechem)
mpse <- MPSE(assays=list(Abundance=t(varespec)), colData=varechem)
mpse
mpse %<>%
    mp_cal_cca(.abundance=Abundance,
               .formula=~Al + P*(K + Baresoil),
               action="add")
mpse
mpse %>% mp_plot_ord(.ord=CCA, .group=Al, .size=K, show.sample=FALSE, bg.colour="black", colour="white")
```

---

mp_cal_clust *Hierarchical cluster analysis for the samples with MPSE or tbl_mpse object*

Description

Hierarchical cluster analysis for the samples with MPSE or tbl_mpse object

Usage

```
mp_cal_clust(
  .data,
  .abundance,
  distmethod = "bray",
  hclustmethod = "average",
  action = "get",
  ...
)

## S4 method for signature 'MPSE'
mp_cal_clust(
  .data,
```

```
  .abundance,
  distmethod = "bray",
  hclustmethod = "average",
  action = "get",
  ...
)

## S4 method for signature 'tbl_mpse'
mp_cal_clust(
  .data,
  .abundance,
  distmethod = "bray",
  hclustmethod = "average",
  action = "get",
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_cal_clust(
  .data,
  .abundance,
  distmethod = "bray",
  hclustmethod = "average",
  action = "get",
  ...
)
```

Arguments

.data	the MPSE or tbl_mpse object
.abundance	the name of abundance to be calculated.
distmethod	the method of distance.
hclustmethod	the method of hierarchical cluster
action	a character "add" will return a MPSE object with the cluster result as a attributes, and it can be extracted with 'object "only" or "get" will return 'treedata' object, default is 'get'.
...	additional parameters

Value

update object with the action argument, the treedata object contained hierarchical cluster analysis of sample, it can be visualized with 'ggtree' directly.

Author(s)

Shuangbin Xu

Examples

```
library(ggtree)
library(ggplot2)
data(mouse.time.mpse)
res <- mouse.time.mpse %>%
 mp_decostand(.abundance=Abundance) %>%
 mp_cal_clust(.abundance=hellinger, distmethod="bray")
res
res %>%
 ggtree() +
 geom_tippoint(aes(color=time))
```

---

mp_cal_dca *Detrended Correspondence Analysis with MPSE or tbl_mpse object*

Description

Detrended Correspondence Analysis with MPSE or tbl_mpse object

Usage

```
mp_cal_dca(.data, .abundance, .dim = 3, action = "add", origin = TRUE, ...)

## S4 method for signature 'MPSE'
mp_cal_dca(.data, .abundance, .dim = 3, action = "add", origin = TRUE, ...)

## S4 method for signature 'tbl_mpse'
mp_cal_dca(.data, .abundance, .dim = 3, action = "add", origin = TRUE, ...)

## S4 method for signature 'grouped_df_mpse'
mp_cal_dca(.data, .abundance, .dim = 3, action = "add", origin = TRUE, ...)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of abundance to be calculated.
.dim	integer The number of dimensions to be returned, default is 3.
action	character "add" joins the 'decorana' result to the object, "only" return a non-redundant tibble with the 'decorana' result. "get" return 'decorana' object can be processed with related vegan function.
origin	logical Use true origin even in detrended correspondence analysis. default is TRUE.
...	additional parameters see also 'vegan::decorana'

Value

update object or tbl according to the action.

mp_cal_dist *Calculate the distances between the samples or features with specified abundance.*

Description

Calculate the distances between the samples or features with specified abundance.

Usage

```
mp_cal_dist(
  .data,
  .abundance,
  .env = NULL,
  distmethod = "bray",
  action = "add",
  scale = FALSE,
  cal.feature.dist = FALSE,
  ...
)

## S4 method for signature 'MPSE'
mp_cal_dist(
  .data,
  .abundance,
  .env = NULL,
  distmethod = "bray",
  action = "add",
  scale = FALSE,
  cal.feature.dist = FALSE,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_cal_dist(
  .data,
  .abundance,
  .env = NULL,
  distmethod = "bray",
  action = "add",
  scale = FALSE,
  cal.feature.dist = FALSE,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_cal_dist(
```

```
  .data,
  .abundance,
  .env = NULL,
  distmethod = "bray",
  action = "add",
  scale = FALSE,
  cal.feature.dist = FALSE,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of otu abundance to be calculated
.env	the column names of continuous environment factors, default is NULL.
distmethod	character the method to calculate distance. option is "manhattan", "euclidean", "canberra", "bray", "kulczynski", "jaccard", "gower", "altGower", "morisita", "horn", "mountford", "raup", "binomial", "chao", "cao", "mahalanobis", "chisq", "chord", "aitchison", "robust.aitchison" (implemented in vegdist of vegan), and "w", "-1", "c", "wb", "r", "I", "e", "t", "me", "j", "sor", "m", "-2", "co", "cc", "g", "-3", "l", "19", "hk", "rlb", "sim", "gl", "z" (implemented in betadiver of vegan), "maximum", "binary", "minkowski" (implemented in dist of stats), "unifrac", "weighted unifrac" (implemented in phyloseq), "cor", "abscor", "cosangle", "abscosangle" (implemented in hopach), or other customized distance function.
action	character, "add" joins the distance data to the object, "only" return a non-redundant tibble with the distance information. "get" return 'dist' object.
scale	logical whether scale the metric of environment (.env is provided) before the distance was calculated, default is FALSE. The environment matrix can be processed when it was joined to the MPSE or tbl_mpse object.
cal.feature.dist	logical whether to calculate the distance between the features. default is FALSE, meaning calculate the distance between the samples.
...	additional parameters. some dot arguments if distmethod is unifrac or weighted unifrac: • weighted logical, whether to use weighted-UniFrac calculation, which considers the relative abundance of taxa, default is FALSE, meaning unweightrd-UniFrac, which only considers presence/absence of taxa. • normalized logical, whether normaized the branch length of tree to the range between 0 and 1 when the weighted=TRUE. • parallel logical, whether to execute the calculation in parallel, default is FALSE.

Value

update object or tibble according the 'action'

Author(s)

Shuangbin Xu

See Also

[mp_extract_dist()] and [mp_plot_dist()]

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %<>%
    mp_decostand(.abundance=Abundance) %>%
    mp_cal_dist(.abundance=hellinger, distmethod="bray")
mouse.time.mpse
p1 <- mouse.time.mpse %>%
        mp_plot_dist(.distmethod = bray)
p2 <- mouse.time.mpse %>%
        mp_plot_dist(.distmethod = bray, .group = time, group.test = TRUE)
p3 <- mouse.time.mpse %>%
        mp_plot_dist(.distmethod = bray, .group = time)
# adjust the legend of heatmap of distance between the samples.
# the p3 is a aplot object, we define set_scale_theme to adjust the
# character (color, size or legend size) of figure with specified
# 'aes_var' according to legend title.
library(ggplot2)
p3 %>%
   set_scale_theme(
     x = scale_size_continuous(
       range = c(0.1, 4),
       guide = guide_legend(keywidth = 0.5, keyheight = 1)),
     aes_var = bray
   ) %>%
   set_scale_theme(
     x = scale_colour_gradient(
       guide = guide_legend(keywidth = 0.5, keyheight = 1)),
     aes_var = bray
   ) %>%
   set_scale_theme(
     x = scale_fill_manual(values = c("orangered", "deepskyblue"),
    guide = guide_legend(keywidth = 0.5, keyheight = 0.5, label.theme = element_text(size=6))),
     aes_var = time) %>%
   set_scale_theme(
     x = theme(axis.text=element_text(size=6), panel.background=element_blank()),
     aes_var = bray
   )
## Not run:
# Visualization manual
library(ggplot2)
tbl <- mouse.time.mpse %>%
       mp_extract_dist(distmethod="bray", .group=time)
tbl
tbl %>%
```

```
  ggplot(aes(x=GroupsComparison, y=bray)) +
  geom_boxplot(aes(fill=GroupsComparison)) +
  geom_jitter(width=0.1) +
  xlab(NULL) +
  theme(legend.position="none")

## End(Not run)
```

`mp_cal_divergence` *calculate the divergence with MPSE or tbl_mpse*

Description

calculate the divergence with MPSE or tbl_mpse

Usage

```
mp_cal_divergence(
  .data,
  .abundance,
  .name = "divergence",
  reference = "mean",
  distFUN = vegan::vegdist,
  method = "bray",
  action = "add",
  ...
)

## S4 method for signature 'MPSE'
mp_cal_divergence(
  .data,
  .abundance,
  .name = "divergence",
  reference = "mean",
  distFUN = vegan::vegdist,
  method = "bray",
  action = "add",
  ...
)

## S4 method for signature 'tbl_mpse'
mp_cal_divergence(
  .data,
  .abundance,
  .name = "divergence",
  reference = "mean",
  distFUN = vegan::vegdist,
```

```
  method = "bray",
  action = "add",
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_cal_divergence(
  .data,
  .abundance,
  .name = "divergence",
  reference = "mean",
  distFUN = vegan::vegdist,
  method = "bray",
  action = "add",
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	The column name of OTU abundance column to be calculate.
.name	the colname name of the divergence results, default is 'divergence'.
reference	a no-empty character, either 'median' or 'mean' or the sample name, or a numeric vector which has length equal to the number of features, default is 'mean'.
distFUN	the function to calculate the distance between the reference and samples, default is 'vegan::vegdist'.
method	the method to calculate the distance, which will pass to the function that is specified in 'distFUN', default is 'bray'.
action	character it has three options, "add" joins the new information to the input tbl (default), "only" return a non-redundant tibble with the just new information, ang 'get' return a 'alphasample' object.
...	additional arguments, see also the arguments of 'distFUN' function.

Value

update object or other (refer to action)

Author(s)

Shuangbin Xu

See Also

[mp_plot_alpha()]

Examples

```
## Not run:
# example(mp_cal_divergence, run.dontrun = TRUE) to run the example.
data(mouse.time.mpse)
mouse.time.mpse %>%
  mp_cal_divergence(
    .abundance = Abundance,
    .name = 'divergence.mean',
    distFUN = vegan::vegdist,
    method = 'bray'
  ) %>%
  mp_plot_alpha(
    .alpha = divergence.mean,
    .group = time,
  )

## End(Not run)
```

`mp_cal_nmds` *Nonmetric Multidimensional Scaling Analysis with MPSE or tbl_mpse object*

Description

Nonmetric Multidimensional Scaling Analysis with MPSE or tbl_mpse object

Usage

```
mp_cal_nmds(
  .data,
  .abundance,
  distmethod = "bray",
  .dim = 2,
  action = "add",
  seed = 123,
  ...
)

## S4 method for signature 'MPSE'
mp_cal_nmds(
  .data,
  .abundance,
  distmethod = "bray",
  .dim = 2,
  action = "add",
  seed = 123,
  ...
)
```

```
## S4 method for signature 'tbl_mpse'
mp_cal_nmds(
  .data,
  .abundance,
  distmethod = "bray",
  .dim = 2,
  action = "add",
  seed = 123,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_cal_nmds(
  .data,
  .abundance,
  distmethod = "bray",
  .dim = 2,
  action = "add",
  seed = 123,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of abundance to be calculated.
distmethod	character the method to calculate distance.
.dim	integer The number of dimensions to be returned, default is 2.
action	character "add" joins the NMDS result to the object, "only" return a non-redundant tibble with the NMDS result. "get" return 'metaMDS' object can be analyzed with related 'vegan' function.
seed	a random seed to make this analysis reproducible, default is 123.
...	additional parameters see also 'mp_cal_dist'.

Value

update object or tbl according to the action.

Author(s)

Shuangbin Xu

Examples

```
data(mouse.time.mpse)
mpse <- mouse.time.mpse %>%
        mp_decostand(.abundance=Abundance) %>%
```

```
        mp_cal_nmds(.abundance=hellinger, distmethod="bray", action="add")
library(ggplot2)
p <- mpse %>% mp_plot_ord(.ord=nmds,
                          .group=time,
                          .color=time,
                          .alpha=0.8,
                          ellipse=TRUE,
                          show.sample=TRUE)
p <- p +
     scale_fill_manual(values=c("#00AED7", "#009E73")) +
     scale_color_manual(values=c("#00AED7", "#009E73"))
## Not run:
mouse.time.mpse %>%
  mp_decostand(.abundance=Abundance) %>%
  mp_cal_nmds(.abundance=hellinger, distmethod="bray", .dim=2, action="only") -> tbl
tbl
x <- names(tbl)[grepl("NMDS1", names(tbl))] %>% as.symbol()
y <- names(tbl)[grepl("NMDS2", names(tbl))] %>% as.symbol()
library(ggplot2)
tbl %>%
 ggplot(aes(x=!!x, y=!!y, color=time)) +
 geom_point() +
 geom_vline(xintercept=0, color="grey20", linetype=2) +
 geom_hline(yintercept=0, color="grey20", linetype=2) +
 theme_bw() +
 theme(panel.grid=element_blank())

## End(Not run)
```

---

mp_cal_pca *Principal Components Analysis with MPSE or tbl_mpse object*

Description

Principal Components Analysis with MPSE or tbl_mpse object

Usage

```
mp_cal_pca(.data, .abundance, .dim = 3, action = "add", ...)

## S4 method for signature 'MPSE'
mp_cal_pca(.data, .abundance, .dim = 3, action = "add", ...)

## S4 method for signature 'tbl_mpse'
mp_cal_pca(.data, .abundance, .dim = 3, action = "add", ...)

## S4 method for signature 'grouped_df_mpse'
mp_cal_pca(.data, .abundance, .dim = 3, action = "add", ...)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of abundance to be calculated.
.dim	integer The number of dimensions to be returned, default is 3.
action	character "add" joins the pca result to the object, "only" return a non-redundant tibble with the pca result. "get" return 'prcomp' object.
...	additional parameters see also 'prcomp'

Value

update object or tbl according to the action.

Author(s)

Shuangbin Xu

Examples

```
data(mouse.time.mpse)
library(ggplot2)
mpse <- mouse.time.mpse %>%
          mp_decostand(.abundance=Abundance) %>%
          mp_cal_pca(.abundance=hellinger, action="add")
mpse
p1 <- mpse %>% mp_plot_ord(.ord=pca, .group=time, ellipse=TRUE)
p2 <- mpse %>% mp_plot_ord(.ord=pca, .group=time, .color=time, ellipse=TRUE)
p1 + scale_fill_manual(values=c("#00AED7", "#009E73"))
p2 + scale_fill_manual(values=c("#00AED7", "#009E73")) +
     scale_color_manual(values=c("#00AED7", "#009E73"))
## Not run:
# action = "only" to extract the non-redundant tibble to visualize
tbl <- mouse.time.mpse %>%
          mp_decostand(.abundance=Abundance) %>%
          mp_cal_pca(.abundance=hellinger, action="only")
tbl
x <- names(tbl)[grepl("PC1 ", names(tbl))] %>% as.symbol()
y <- names(tbl)[grepl("PC2 ", names(tbl))] %>% as.symbol()
ggplot(tbl) +
 geom_point(aes(x=!!x, y=!!y, color=time))

## End(Not run)
```

`mp_cal_pcoa` *Principal Coordinate Analysis with MPSE or tbl_mpse object*

Description

Principal Coordinate Analysis with MPSE or tbl_mpse object

Usage

```
mp_cal_pcoa(
  .data,
  .abundance,
  distmethod = "bray",
  .dim = 3,
  action = "add",
  ...
)

## S4 method for signature 'MPSE'
mp_cal_pcoa(
  .data,
  .abundance,
  distmethod = "bray",
  .dim = 3,
  action = "add",
  ...
)

## S4 method for signature 'tbl_mpse'
mp_cal_pcoa(
  .data,
  .abundance,
  distmethod = "bray",
  .dim = 3,
  action = "add",
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_cal_pcoa(
  .data,
  .abundance,
  distmethod = "bray",
  .dim = 3,
  action = "add",
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of abundance to be calculated.
distmethod	character the method to calculate distance.
.dim	integer The number of dimensions to be returned, default is 3.
action	character "add" joins the pca result to the object and the 'pcoa' object also was add to the internal attributes of the object, "only" return a non-redundant tibble with the pca result. "get" return 'pcoa' object.
...	additional parameters see also 'mp_cal_dist'.

Value

update object or tbl according to the action.

Author(s)

Shuangbin Xu

Examples

```
data(mouse.time.mpse)
mpse <- mouse.time.mpse %>%
        mp_decostand(.abundance=Abundance)
mpse
mpse %<>% mp_cal_pcoa(.abundance=hellinger, stmethod="bray", action="add")
library(ggplot2)
p <- mpse %>% mp_plot_ord(.ord=pcoa, .group=time, .color=time, ellipse=TRUE)
p <- p +
     scale_fill_manual(values=c("#00AED7", "#009E73")) +
     scale_color_manual(values=c("#00AED7", "#009E73"))
## Not run:
# Or run with action='only' and return tbl_df to visualize manual.
mouse.time.mpse %>%
  mp_decostand(.abundance=Abundance) %>%
  mp_cal_pcoa(.abundance=hellinger, distmethod="bray", .dim=2, action="only") -> tbl
tbl
x <- names(tbl)[grepl("PCo1 ", names(tbl))] %>% as.symbol()
y <- names(tbl)[grepl("PCo2 ", names(tbl))] %>% as.symbol()
library(ggplot2)
tbl %>%
 ggplot(aes(x=!!x, y=!!y, color=time)) +
 stat_ellipse(aes(fill=time), geom="polygon", alpha=0.5) +
 geom_point() +
 geom_vline(xintercept=0, color="grey20", linetype=2) +
 geom_hline(yintercept=0, color="grey20", linetype=2) +
 theme_bw() +
 theme(panel.grid=element_blank())

## End(Not run)
```

`mp_cal_pd_metric` *Calculating related phylogenetic alpha metric with MPSE or tbl_mpse object*

Description

Calculating related phylogenetic alpha metric with MPSE or tbl_mpse object

Usage

```
mp_cal_pd_metric(
  .data,
  .abundance,
  action = "add",
  metric = c("PAE", "NRI", "NTI", "PD", "HAED", "EAED", "all"),
  abundance.weighted = FALSE,
  force = FALSE,
  seed = 123,
  ...
)

## S4 method for signature 'MPSE'
mp_cal_pd_metric(
  .data,
  .abundance,
  action = "add",
  metric = c("PAE", "NRI", "NTI", "PD", "HAED", "EAED", "IAC", "all"),
  abundance.weighted = FALSE,
  force = FALSE,
  seed = 123,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_cal_pd_metric(
  .data,
  .abundance,
  action = "add",
  metric = c("PAE", "NRI", "NTI", "PD", "HAED", "EAED", "all"),
  abundance.weighted = TRUE,
  force = FALSE,
  seed = 123,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_cal_pd_metric(
```

```
  .data,
  .abundance,
  action = "add",
  metric = c("PAE", "NRI", "NTI", "PD", "HAED", "EAED", "all"),
  abundance.weighted = TRUE,
  force = FALSE,
  seed = 123,
  ...
)
```

Arguments

.data	object, MPSE or tbl_mpse object
.abundance	The column name of OTU abundance column to be calculate.
action	character it has three options, "add" joins the new information to the input tbl (default), "only" return a non-redundant tibble with the just new information, ang 'get' return a 'alphasample' object.
metric	the related phylogenetic metric, options is 'NRI', 'NTI', 'PD', 'PAE', 'HAED', 'EAED', 'IAC', 'all', default is 'PAE', 'all' meaning all the metrics ('NRI', 'NTI', 'PD', 'PAE', 'HAED', 'EAED', 'IAC').
abundance.weighted	logical, whether calculate mean nearest taxon distances for each species weighted by species abundance, default is TRUE.
force	logical whether calculate the alpha index even the '.abundance' is not rarefied, default is FALSE.
seed	integer a random seed to make the result reproducible, default is 123.
...	additional arguments see also "ses.mpd" and "ses.mntd" of "picante".

Value

update object.

Author(s)

Shuangbin Xu

References

Cadotte, M.W., Jonathan Davies, T., Regetz, J., Kembel, S.W., Cleland, E. and Oakley, T.H. (2010), Phylogenetic diversity metrics for ecological communities: integrating species richness, abundance and evolutionary history. Ecology Letters, 13: 96-105. https://doi.org/10.1111/j.1461-0248.2009.01405.x.

Webb, C. O. (2000). Exploring the phylogenetic structure of ecological communities: an example for rain forest trees. The American Naturalist, 156(2), 145-155. https://doi.org/10.1086/303378.

Examples

```
## Not run:
  suppressPackageStartupMessages(library(curatedMetagenomicData))
  xx <- curatedMetagenomicData('ZellerG_2014.relative_abundance', dryrun=F)
  xx[[1]] %>% as.mpse -> mpse
  mpse %<>%
    mp_cal_pd_metric(
      .abundance = Abundance,
      force = TRUE,
      metric = 'PAE'
    )
  mpse %>%
    mp_plot_alpha(
      .alpha = PAE,
      .group = disease
  )

## End(Not run)
```

---

mp_cal_rarecurve *Calculating the different alpha diversities index with different depth*

Description

Calculating the different alpha diversities index with different depth

Usage

```
mp_cal_rarecurve(
  .data,
  .abundance = NULL,
  action = "add",
  chunks = 400,
  seed = 123,
  force = FALSE,
  ...
)

## S4 method for signature 'MPSE'
mp_cal_rarecurve(
  .data,
  .abundance = NULL,
  action = "add",
  chunks = 400,
  seed = 123,
  force = FALSE,
  ...
)
```

```
## S4 method for signature 'tbl_mpse'
mp_cal_rarecurve(
  .data,
  .abundance = NULL,
  action = "add",
  chunks = 400,
  seed = 123,
  force = FALSE,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_cal_rarecurve(
  .data,
  .abundance = NULL,
  action = "add",
  chunks = 400,
  seed = 123,
  force = FALSE,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of otu abundance to be calculated.
action	character it has three options, "add" joins the new information to the input tbl (default), "only" return a non-redundant tibble with the just new information, ang 'get' return a 'rarecurve' object.
chunks	numeric the split number of each sample to calculate alpha diversity, default is 400. eg. A sample has total 40000 reads, if chunks is 400, it will be split to 100 sub-samples (100, 200, 300,..., 40000), then alpha diversity index was calculated based on the sub-samples.
seed	a random seed to make the result reproducible, default is 123.
force	logical whether calculate rarecurve forcibly when the '.abundance' is not be rarefied, default is FALSE
...	additional parameters.

Value

update rarecurce calss

Author(s)

Shuangbin Xu

See Also

[mp_plot_rarecurve()] and [mp_extract_rarecurve()]

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %>%
mp_rrarefy() -> mpse
mpse
# larger 'chunks' means more robust, but it will become slower.
mpse %<>% mp_cal_rarecurve(.abundance=RareAbundance, chunks=100, action="add")
mpse
p1 <- mpse %>%
      mp_plot_rarecurve(.rare=RareAbundanceRarecurve, .alpha="Observe")
p2 <- mpse %>%
      mp_plot_rarecurve(.rare=RareAbundanceRarecurve, .alpha=c("Observe", "ACE"))
```

---

mp_cal_rda *[Partial] [Constrained] Redundancy Analysis with MPSE or tbl_mpse object*

Description

[Partial] [Constrained] Redundancy Analysis with MPSE or tbl_mpse object

Usage

```
mp_cal_rda(.data, .abundance, .formula = NULL, .dim = 3, action = "add", ...)

## S4 method for signature 'MPSE'
mp_cal_rda(.data, .abundance, .formula = NULL, .dim = 3, action = "add", ...)

## S4 method for signature 'tbl_mpse'
mp_cal_rda(.data, .abundance, .formula = NULL, .dim = 3, action = "add", ...)

## S4 method for signature 'grouped_df_mpse'
mp_cal_rda(.data, .abundance, .formula = NULL, .dim = 3, action = "add", ...)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of abundance to be calculated.
.formula	Model formula right hand side gives the constraining variables, and conditioning variables can be given within a special function 'Condition' and keep left empty, such as ~ A + B or ~ A + Condition(B), default is NULL.
.dim	integer The number of dimensions to be returned, default is 3.

action	character "add" joins the rda result to the object, "only" return a non-redundant tibble with the rda result. "get" return 'rda' object can be analyzed using the related vegan funtion.
...	additional parameters see also 'rda' of vegan.

Value

update object according action argument

Author(s)

Shuangbin Xu

Examples

```
library(vegan)
data(varespec, varechem)
mpse <- MPSE(assays=list(Abundance=t(varespec)), colData=varechem)
mpse
mpse %>%
  mp_cal_rda(.abundance=Abundance,
             .formula=~Al + P*(K + Baresoil),
             .dim = 3,
             action="add") %>%
  mp_plot_ord(show.sample=TRUE)
```

---

mp_cal_upset *Calculating the samples or groups for each OTU, the result can be visualized by 'ggupset'*

Description

Calculating the samples or groups for each OTU, the result can be visualized by 'ggupset'

Usage

```
mp_cal_upset(
  .data,
  .group,
  .abundance = NULL,
  action = "add",
  force = FALSE,
  ...
)

## S4 method for signature 'MPSE'
mp_cal_upset(
  .data,
```

```
  .group,
  .abundance = NULL,
  action = "add",
  force = FALSE,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_cal_upset(
  .data,
  .group,
  .abundance = NULL,
  action = "add",
  force = FALSE,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_cal_upset(
  .data,
  .group,
  .abundance = NULL,
  action = "add",
  force = FALSE,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.group	the name of group to be calculated. if it is no provided, the sample will be used.
.abundance	the name of otu abundance to be calculated. if it is null, the rarefied abundance will be used.
action	character, "add" joins the new information to the tibble of tbl_mpse or rowData of MPSE. "only" and "get" return a non-redundant tibble with the just new information. which is a treedata object.
force	logical whether calculate the relative abundance forcibly when the abundance is not be rarefied, default is FALSE.
...	additional parameters.

Value

update object or tibble according the 'action'

Author(s)

Shuangbin Xu

See Also

[mp_plot_upset()]

Examples

```
data(mouse.time.mpse)
mpse <- mouse.time.mpse %>%
        mp_rrarefy() %>%
        mp_cal_upset(.abundance=RareAbundance, .group=time, action="add")
mpse
library(ggplot2)
library(ggupset)
p <- mpse %>% mp_plot_upset(.group=time, .upset=ggupsetOftime)
p
# or set action="only"
## Not run:
tbl <- mouse.time.mpse %>%
       mp_rrarefy() %>%
       mp_cal_upset(.abundance=RareAbundance, .group=time, action="only")
tbl
p2 <- tbl %>%
      ggplot(aes(x=ggupsetOftime)) +
      geom_bar() +
      ggupset::scale_x_upset() +
      ggupset::theme_combmatrix(combmatrix.label.extra_spacing=30)

## End(Not run)
```

---

mp_cal_venn *Calculating the OTU for each sample or group, the result can be visualized by 'ggVennDiagram'*

Description

Calculating the OTU for each sample or group, the result can be visualized by 'ggVennDiagram'

Usage

```
mp_cal_venn(
  .data,
  .group,
  .abundance = NULL,
  action = "add",
  force = FALSE,
  ...
)

## S4 method for signature 'MPSE'
```

```
mp_cal_venn(
  .data,
  .group,
  .abundance = NULL,
  action = "add",
  force = FALSE,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_cal_venn(
  .data,
  .group,
  .abundance = NULL,
  action = "add",
  force = FALSE,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_cal_venn(
  .data,
  .group,
  .abundance = NULL,
  action = "add",
  force = FALSE,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.group	the name of group to be calculated. if it is no provided, the sample will be used.
.abundance	the name of otu abundance to be calculated. if it is null, the rarefied abundance will be used.
action	character, "add" joins the new information to the tibble of tbl_mpse or rowData of MPSE. "only" and "get" return a non-redundant tibble with the just new information.
force	logical whether calculate the relative abundance forcibly when the abundance is not be rarefied, default is FALSE.
...	additional parameters.

Value

update object or tibble according the 'action'

Author(s)

Shuangbin Xu

See Also

[mp_plot_venn()]

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %>%
mp_rrarefy() %>%
mp_cal_venn(.abundance=RareAbundance, .group=time, action="add") -> mpse
mpse
p <- mpse %>% mp_plot_venn(.venn = vennOftime, .group = time)
## Not run:
# visualized by manual
library(ggplot2)
mpse %>%
  mp_extract_sample() %>%
  select(time, vennOftime) %>%
  distinct() %>%
  pull(var=vennOftime, name=time) %>%
  ggVennDiagram::ggVennDiagram()

## End(Not run)
```

---

mp_decostand *This Function Provideds Several Standardization Methods for Community Data*

Description

This Function Provideds Several Standardization Methods for Community Data

Usage

```
mp_decostand(.data, .abundance = NULL, method = "hellinger", logbase = 2, ...)

## S4 method for signature 'data.frame'
mp_decostand(.data, .abundance = NULL, method = "hellinger", logbase = 2, ...)

## S4 method for signature 'MPSE'
mp_decostand(.data, .abundance = NULL, method = "hellinger", logbase = 2, ...)

## S4 method for signature 'tbl_mpse'
mp_decostand(.data, .abundance = NULL, method = "hellinger", logbase = 2, ...)

## S4 method for signature 'grouped_df_mpse'
mp_decostand(.data, .abundance = NULL, method = "hellinger", logbase = 2, ...)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the names of otu abundance to be applied standardization.
method	character the name of standardization method, it can one of 'total', 'max', 'frequency', 'normalize', 'range', 'rank', 'rrank', 'standardize' 'pa', 'chi.square', 'hellinger' and 'log', see also decostand
logbase	numeric The logarithm base used in 'method=log', default is 2.
...	additional parameters, see also decostand

Value

update object

Author(s)

Shuangbin Xu

Source

mp_decostand for data.frame object is a wrapper method of vegan::decostand from the vegan package

See Also

[mp_extract_assays()] and [mp_rrarefy()]

decostand

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %>%
mp_decostand(.abundance=Abundance, method="hellinger")
```

---

mp_diff_analysis *Differential expression analysis for MPSE or tbl_mpse object*

Description

Differential expression analysis for MPSE or tbl_mpse object

Usage

```
mp_diff_analysis(
  .data,
  .abundance,
  .group,
  .sec.group = NULL,
  action = "add",
  tip.level = "OTU",
  force = FALSE,
  relative = TRUE,
  taxa.class = "all",
  first.test.method = "kruskal.test",
  first.test.alpha = 0.05,
  p.adjust = "fdr",
  filter.p = "fdr",
  strict = TRUE,
  fc.method = "generalizedFC",
  second.test.method = "wilcox.test",
  second.test.alpha = 0.05,
  cl.min = 5,
  cl.test = TRUE,
  subcl.min = 3,
  subcl.test = TRUE,
  ml.method = "lda",
  normalization = 1e+06,
  ldascore = 2,
  bootnums = 30,
  sample.prop.boot = 0.7,
  ci = 0.95,
  seed = 123,
  type = "species",
  ...
)

## S4 method for signature 'MPSE'
mp_diff_analysis(
  .data,
  .abundance,
  .group,
  .sec.group = NULL,
  action = "add",
  tip.level = "OTU",
  force = FALSE,
  relative = TRUE,
  taxa.class = "all",
  first.test.method = "kruskal.test",
  first.test.alpha = 0.05,
  p.adjust = "fdr",
```

```
  filter.p = "fdr",
  strict = TRUE,
  fc.method = "generalizedFC",
  second.test.method = "wilcox.test",
  second.test.alpha = 0.05,
  cl.min = 5,
  cl.test = TRUE,
  subcl.min = 3,
  subcl.test = TRUE,
  ml.method = "lda",
  normalization = 1e+06,
  ldascore = 2,
  bootnums = 30,
  sample.prop.boot = 0.7,
  ci = 0.95,
  seed = 123,
  type = "species",
  ...
)

## S4 method for signature 'tbl_mpse'
mp_diff_analysis(
  .data,
  .abundance,
  .group,
  .sec.group = NULL,
  action = "add",
  tip.level = "OTU",
  force = FALSE,
  relative = TRUE,
  taxa.class = "all",
  first.test.method = "kruskal.test",
  first.test.alpha = 0.05,
  p.adjust = "fdr",
  filter.p = "fdr",
  strict = TRUE,
  fc.method = "generalizedFC",
  second.test.method = "wilcox.test",
  second.test.alpha = 0.05,
  cl.min = 5,
  cl.test = TRUE,
  subcl.min = 3,
  subcl.test = TRUE,
  ml.method = "lda",
  normalization = 1e+06,
  ldascore = 2,
  bootnums = 30,
  sample.prop.boot = 0.7,
```

```
  ci = 0.95,
  seed = 123,
  type = "species",
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_diff_analysis(
  .data,
  .abundance,
  .group,
  .sec.group = NULL,
  action = "add",
  tip.level = "OTU",
  force = FALSE,
  relative = TRUE,
  taxa.class = "all",
  first.test.method = "kruskal.test",
  first.test.alpha = 0.05,
  p.adjust = "fdr",
  filter.p = "fdr",
  strict = TRUE,
  fc.method = "generalizedFC",
  second.test.method = "wilcox.test",
  second.test.alpha = 0.05,
  cl.min = 5,
  cl.test = TRUE,
  subcl.min = 3,
  subcl.test = TRUE,
  ml.method = "lda",
  normalization = 1e+06,
  ldascore = 2,
  bootnums = 30,
  sample.prop.boot = 0.7,
  ci = 0.95,
  seed = 123,
  type = "species",
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of abundance to be calculated
.group	the group name of the samples to be calculated.
.sec.group	the second group name of the samples to be calculated.
action	character, "add" joins the new information to the taxatree (if it exists) or rowData and return MPSE object,"only" return a non-redundant tibble with the result of

different analysis. "get" return 'diffAnalysisClass' object.

`tip.level` character the taxa level to be as tip level

`force` logical whether to calculate the relative abundance forcibly when the abundance is not be rarefied, default is FALSE.

`relative` logical whether calculate the relative abundance.

`taxa.class` character if taxa class is not 'all', only the specified taxa class will be identified, default is 'all'.

`first.test.method`
the method for first test, option is "kruskal.test", "oneway.test", "lm", "glm", or "glm.nb", "kruskal_test", "oneway_test" of "coin" package. default is "kruskal.test".

`first.test.alpha`
numeric the alpha value for the first test, default is 0.05.

`p.adjust` character the correction method, default is "fdr", see also p.adjust function default is fdr.

`filter.p` character the method to filter pvalue, default is fdr, meanings the features that fdr <= .first.test.alpha will be kept, if it is set to pvalue, meanings the features that pvalue <= .first.test.alpha will be kept.

`strict` logical whether to performed in one-against-one when .sec.group is provided, default is TRUE (strict).

`fc.method` character the method to check which group has more abundance for the significantly different features, default is "generalizedFC", options are `generalizedFC`, `compare_median`, `compare_mean`.

`second.test.method`
the method for one-against-one (the second test), default is "wilcox.test" other option is one of 'wilcox_test' of 'coin'; 'glm'; 'glm.nb' of 'MASS'.

`second.test.alpha`
numeric the alpha value for the second test, default is 0.05.

`cl.min` integer the minimum number of samples per group for performing test, default is 5.

`cl.test` logical whether to perform test (second test) between the groups (the number of sample of the .group should be also larger that cl.min), default is TRUE.

`subcl.min` integer the minimum number of samples in each second groups for performing test, default is 3.

`subcl.test` logical whether to perform test for between the second groups (the .sec.group should be provided and the number sample of each .sec.group should be larger than subcl.min, and strict is TRUE), default is TRUE.

`ml.method` the method for calculating the effect size of features, option is 'lda' or 'rf'. default is 'lda'.

`normalization` integer set a big number if to get more meaningful values for the LDA score, or you can set NULL for no normalization, default is 1000000.

`ldascore` numeric the threshold on the absolute value of the logarithmic LDA score, default is 2.

`bootnums` integer, set the number of bootstrap iteration for lda or rf, default is 30.

sample.prop.boot
: numeric range from 0 to 1, the proportion of samples for calculating the effect size of features, default is 0.7.

ci
: numeric, the confidence interval of effect size (LDA or MDA), default is 0.95.

seed
: a random seed to make the analysis reproducible, default is 123.

type
: character type="species" meaning the abundance matrix is from the species abundance, other option is "others", default is "species".

...
: additional parameters

Value

update object according to the action argument.

Author(s)

Shuangbin Xu

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %<>%
  mp_rrarefy()
mouse.time.mpse
mouse.time.mpse %<>%
  mp_diff_analysis(.abundance=RareAbundance,
                   .group=time,
                   first.test.alpha=0.01,
                   action="add")
library(ggplot2)
p <- mouse.time.mpse %>% mp_plot_diff_res()
flag <- packageVersion("ggnewscale") >= "0.5.0"
# if flag is TRUE, you can also use p$ggnewscale to view the renamed scales.
new.fill <- ifelse(flag , "fill_ggnewscale_2", "fill_new")
p <- p +
     scale_fill_manual(
     aesthetics = new.fill, # The fill aes was renamed to `new.fill` for the abundance dotplot layer
       values = c("skyblue", "orange")
     )  +
     scale_fill_manual(
       values=c("skyblue", "orange") # The LDA barplot layer
     )
### and the fill aes for hight light layer of tree was renamed to `new.fill2`
### because the layer is the first layer used `fill`
new.fill2 <- ifelse(flag, "fill_ggnewscale_1", "fill_new_new")
p <- p +
     scale_fill_manual(
       aesthetics = new.fill2,
       values = c("#E41A1C", "#377EB8", "#4DAF4A",
                  "#984EA3", "#FF7F00", "#FFFF33",
                  "#A65628", "#F781BF", "#999999")
     )
```

```
p
## Not run:
  ### visualizing the differential taxa with cladogram
  f <- mouse.time.mpse %>%
       mp_plot_diff_cladogram(
         label.size = 2.5,
         hilight.alpha = .3,
         bg.tree.size = .5,
         bg.point.size = 2,
         bg.point.stroke = .25
       ) +
       scale_fill_diff_cladogram(
         values = c('skyblue', 'orange')
       ) +
       scale_size_continuous(range = c(1, 4))
  f

## End(Not run)
```

---

mp_diff_clade *Differential internal and tip nodes (clades) analysis for MPSE or tbl_mpse object*

Description

Differential internal and tip nodes (clades) analysis for MPSE or tbl_mpse object

Usage

```
mp_diff_clade(
  .data,
  .abundance,
  .group,
  .sec.group = NULL,
  action = "add",
  force = FALSE,
  relative = TRUE,
  first.test.method = "kruskal.test",
  first.test.alpha = 0.05,
  p.adjust = "fdr",
  filter.p = "fdr",
  strict = TRUE,
  fc.method = "generalizedFC",
  second.test.method = "wilcox.test",
  second.test.alpha = 0.05,
  cl.min = 5,
  cl.test = TRUE,
  subcl.min = 3,
```

```
  subcl.test = TRUE,
  ml.method = "lda",
  normalization = 1e+06,
  ldascore = 2,
  bootnums = 30,
  sample.prop.boot = 0.7,
  ci = 0.95,
  seed = 123,
  type = "species",
  ...
)

## S4 method for signature 'MPSE'
mp_diff_clade(
  .data,
  .abundance,
  .group,
  .sec.group = NULL,
  action = "add",
  force = FALSE,
  relative = TRUE,
  first.test.method = "kruskal.test",
  first.test.alpha = 0.05,
  p.adjust = "fdr",
  filter.p = "fdr",
  strict = TRUE,
  fc.method = "generalizedFC",
  second.test.method = "wilcox.test",
  second.test.alpha = 0.05,
  cl.min = 5,
  cl.test = TRUE,
  subcl.min = 3,
  subcl.test = TRUE,
  ml.method = "lda",
  normalization = 1e+06,
  ldascore = 2,
  bootnums = 30,
  sample.prop.boot = 0.7,
  ci = 0.95,
  seed = 123,
  type = "species",
  ...
)

## S4 method for signature 'tbl_mpse'
mp_diff_clade(
  .data,
  .abundance,
```

```
  .group,
  .sec.group = NULL,
  action = "add",
  force = FALSE,
  relative = TRUE,
  first.test.method = "kruskal.test",
  first.test.alpha = 0.05,
  p.adjust = "fdr",
  filter.p = "fdr",
  strict = TRUE,
  fc.method = "generalizedFC",
  second.test.method = "wilcox.test",
  second.test.alpha = 0.05,
  cl.min = 5,
  cl.test = TRUE,
  subcl.min = 3,
  subcl.test = TRUE,
  ml.method = "lda",
  normalization = 1e+06,
  ldascore = 2,
  bootnums = 30,
  sample.prop.boot = 0.7,
  ci = 0.95,
  seed = 123,
  type = "species",
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_diff_clade(
  .data,
  .abundance,
  .group,
  .sec.group = NULL,
  action = "add",
  force = FALSE,
  relative = TRUE,
  first.test.method = "kruskal.test",
  first.test.alpha = 0.05,
  p.adjust = "fdr",
  filter.p = "fdr",
  strict = TRUE,
  fc.method = "generalizedFC",
  second.test.method = "wilcox.test",
  second.test.alpha = 0.05,
  cl.min = 5,
  cl.test = TRUE,
  subcl.min = 3,
```

```
  subcl.test = TRUE,
  ml.method = "lda",
  normalization = 1e+06,
  ldascore = 2,
  bootnums = 30,
  sample.prop.boot = 0.7,
  ci = 0.95,
  seed = 123,
  type = "species",
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of abundance to be calculated
.group	the group name of the samples to be calculated.
.sec.group	the second group name of the samples to be calculated.
action	character, "add" joins the new information to the taxatree (if it exists) and otutree (if it exists) or rowData and return MPSE object,"only" return a non-redundant tibble with the result of different analysis. "get" return 'diffAnalysisClass' object.
force	logical whether to calculate the relative abundance forcibly when the abundance is not be rarefied, default is FALSE.
relative	logical whether calculate the relative abundance, default is TRUE.
first.test.method	the method for first test, option is "kruskal.test", "oneway.test", "lm", "glm", or "glm.nb", "kruskal_test", "oneway_test" of "coin" package. default is "kruskal.test".
first.test.alpha	numeric the alpha value for the first test, default is 0.05.
p.adjust	character the correction method, default is "fdr", see also p.adjust function default is fdr.
filter.p	character the method to filter pvalue, default is fdr, meanings the features that fdr <= .first.test.alpha will be kept, if it is set to pvalue, meanings the features that pvalue <= .first.test.alpha will be kept.
strict	logical whether to performed in one-against-one when .sec.group is provided, default is TRUE (strict).
fc.method	character the method to check which group has more abundance for the significantly different features, default is "generalizedFC", options are generalizedFC, compare_median, compare_mean.
second.test.method	the method for one-against-one (the second test), default is "wilcox.test" other option is one of 'wilcox_test' of 'coin'; 'glm'; 'glm.nb' of 'MASS'.
second.test.alpha	numeric the alpha value for the second test, default is 0.05.

cl.min	integer the minimum number of samples per group for performing test, default is 5.
cl.test	logical whether to perform test (second test) between the groups (the number of sample of the .group should be also larger that cl.min), default is TRUE.
subcl.min	integer the minimum number of samples in each second groups for performing test, default is 3.
subcl.test	logical whether to perform test for between the second groups (the .sec.group should be provided and the number sample of each .sec.group should be larger than subcl.min, and strict is TRUE), default is TRUE.
ml.method	the method for calculating the effect size of features, option is 'lda' or 'rf'. default is 'lda'.
normalization	integer set a big number if to get more meaningful values for the LDA score, or you can set NULL for no normalization, default is 1000000.
ldascore	numeric the threshold on the absolute value of the logarithmic LDA score, default is 2.
bootnums	integer, set the number of bootstrap iteration for lda or rf, default is 30.
sample.prop.boot	numeric range from 0 to 1, the proportion of samples for calculating the effect size of features, default is 0.7.
ci	numeric, the confidence interval of effect size (LDA or MDA), default is 0.95.
seed	a random seed to make the analysis reproducible, default is 123.
type	character type="species" meaning the abundance matrix is from the species abundance, other option is "others", default is "species".
...	additional parameters

Value

update object according to the action argument.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
  suppressPackageStartupMessages(library(curatedMetagenomicData))
  xx <- curatedMetagenomicData('ZellerG_2014.relative_abundance', dryrun=F)
  xx[[1]] %>% as.mpse -> mpse
  mpse.agg.clade <- mpse %>%
    mp_aggregate_clade(
      .abundance = Abundance,
      force = TRUE,
      relative = FALSE,
      action = 'add' # other option is 'get' or 'only'.
    )
```

```
  mpse.agg.clade %>% mp_diff_clade(
      .abundance = Abundance,
      force = TRUE,
      relative = FALSE,
      .group = disease,
      fc.method = "compare_mean"
    ) %>%
  mp_extract_otutree() %>%
  dplyr::filter(!is.na(Sign_disease), keep.td = FALSE)

## End(Not run)
```

---

mp_dmn *Fit Dirichlet-Multinomial models to MPSE or tbl_mpse*

Description

Fit Dirichlet-Multinomial models to MPSE or tbl_mpse

Usage

```
mp_dmn(.data, .abundance, k = 1, seed = 123, mc.cores = 2, action = "get", ...)

## S4 method for signature 'MPSE'
mp_dmn(.data, .abundance, k = 1, seed = 123, mc.cores = 2, action = "get", ...)

## S4 method for signature 'tbl_mpse'
mp_dmn(.data, .abundance, k = 1, seed = 123, mc.cores = 2, action = "get", ...)

## S4 method for signature 'grouped_df_mpse'
mp_dmn(.data, .abundance, k = 1, seed = 123, mc.cores = 2, action = "get", ...)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	The column name of OTU abundance column to be calculate.
k	the number of Dirichlet components to fit, default is 1.
seed	random number seed to be reproducible, default is 123.
mc.cores	The number of cores to use, default is 2.
action	character it has three options, 'get' return a 'list' contained DMN (default), "add" joins the new information to the input (can be extracted with mp_extract_internal_attr(name='DMN "only" return a non-redundant tibble with the just new information a column contained 'DMN'.
...	additional parameters, see also the mclapply and dmn.

Value

update object or other (refer to action)

Examples

```
## Not run:
data(mouse.time.mpse)
res <- mouse.time.mpse %>%
       mp_dmn(.abundance = Abundance,
              k = seq_len(2),
              mc.cores = 4,
              action = 'get')
res

## End(Not run)
```

---

mp_dmngroup *Dirichlet-Multinomial generative classifiers to MPSE or tbl_mpse*

Description

Dirichlet-Multinomial generative classifiers to MPSE or tbl_mpse

Usage

```
mp_dmngroup(.data, .abundance, .group, k = 1, action = "get", ...)

## S4 method for signature 'MPSE'
mp_dmngroup(.data, .abundance, .group, k = 1, action = "get", ...)

## S4 method for signature 'tbl_mpse'
mp_dmngroup(.data, .abundance, .group, k = 1, action = "get", ...)

## S4 method for signature 'grouped_df_mpse'
mp_dmngroup(.data, .abundance, .group, k = 1, action = "get", ...)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	The column name of OTU abundance column to be calculate.
.group	the column name of group variable.
k	the number of Dirichlet components to fit, default is 1.
action	character it has three options, 'get' return a 'list' contained DMN (default), "add" joins the new information to the input (can be extracted with mp_extract_internal_attr(name='DMN "only" return a non-redundant tibble with the just new information a column contained 'DMNGroup'.
...	additional parameters, see also the mclapply and dmngroup.

Value

update object or others (refer to action argument)

Examples

```
## Not run:
data(mouse.time.mpse)
mouse.time.mpse %>%
  mp_dmngroup(
    .abundance = Abundance,
    .group = time,
    k=seq_len(2),
    action = 'get'
  )

## End(Not run)
```

---

mp_envfit *Fits an Environmental Vector or Factor onto an Ordination With MPSE or tbl_mpse Object*

Description

Fits an Environmental Vector or Factor onto an Ordination With MPSE or tbl_mpse Object

Usage

```
mp_envfit(
  .data,
  .ord,
  .env,
  .dim = 3,
  action = "only",
  permutations = 999,
  seed = 123,
  ...
)

## S4 method for signature 'MPSE'
mp_envfit(
  .data,
  .ord,
  .env,
  .dim = 3,
  action = "only",
  permutations = 999,
  seed = 123,
  ...
```

```
)

## S4 method for signature 'tbl_mpse'
mp_envfit(
  .data,
  .ord,
  .env,
  .dim = 3,
  action = "only",
  permutations = 999,
  seed = 123,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_envfit(
  .data,
  .ord,
  .env,
  .dim = 3,
  action = "only",
  permutations = 999,
  seed = 123,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.ord	a name of ordination, option it is DCA, NMDS, RDA, CCA.
.env	the names of columns of sample group or environment information.
.dim	integer The number of dimensions to be returned, default is 3.
action	character "add" joins the envfit result to internal attributes of the object, "only" return a non-redundant tibble with the envfit result. "get" return 'envfit' object can be analyzed using the related vegan funtion.
permutations	the number of permutations required, default is 999.
seed	a random seed to make the analysis reproducible, default is 123.
...	additional parameters see also 'vegan::envfit'

Value

update object according action

Author(s)

Shuangbin Xu

Examples

```
library(vegan)
data(varespec, varechem)
mpse <- MPSE(assays=list(Abundance=t(varespec)), colData=varechem)
envformula <- paste("~", paste(colnames(varechem), collapse="+")) %>% as.formula
mpse %<>%
       mp_cal_cca(.abundance=Abundance, .formula=envformula, action="add")
mpse2 <- mpse %>%
         mp_envfit(.ord=cca,
                   .env=colnames(varechem),
                   permutations=9999,
                   action="add")
mpse2 %>% mp_plot_ord(.ord=cca, .group=Al, .size=Mn, show.shample=TRUE, show.envfit=TRUE)
## Not run:
tbl <- mpse %>%
       mp_envfit(.ord=CCA,
                 .env=colnames(varechem),
                 permutations=9999,
                 action="only")
tbl
library(ggplot2)
library(ggrepel)
x <- names(tbl)[grepl("^CCA1 ", names(tbl))] %>% as.symbol()
y <- names(tbl)[grepl("^CCA2 ", names(tbl))] %>% as.symbol()
p <- tbl %>%
     ggplot(aes(x=!!x, y=!!y)) +
     geom_point(aes(color=Al, size=Mn)) +
     geom_segment(data=dr_extract(
                           name="CCA_ENVFIT_tb",
                           .f=td_filter(pvals<=0.05 & label!="Humdepth")
                       ),
                  aes(x=0, y=0, xend=CCA1, yend=CCA2),
                  arrow=arrow(length = unit(0.02, "npc"))
     ) +
     geom_text_repel(data=dr_extract(
                             name="CCA_ENVFIT_tb",
                             .f=td_filter(pvals<=0.05 & label!="Humdepth")
                         ),
                  aes(x=CCA1, y=CCA2, label=label)
     ) +
     geom_vline(xintercept=0, color="grey20", linetype=2) +
     geom_hline(yintercept=0, color="grey20", linetype=2) +
     theme_bw() +
     theme(panel.grid=element_blank())
p

## End(Not run)
```

---

mp_extract_abundance *Extracting the abundance metric from MPSE or tbl_mpse object*

Description

Extracting the abundance metric from the MPSE or tbl_mpse, the 'mp_cal_abundance' must have been run with action='add'.

Usage

```
mp_extract_abundance(x, taxa.class = "all", topn = NULL, rmun = FALSE, ...)

## S4 method for signature 'MPSE'
mp_extract_abundance(x, taxa.class = "all", topn = NULL, rmun = FALSE, ...)

## S4 method for signature 'tbl_mpse'
mp_extract_abundance(x, taxa.class = "all", topn = NULL, rmun = FALSE, ...)

## S4 method for signature 'grouped_df_mpse'
mp_extract_abundance(x, taxa.class = "all", topn = NULL, rmun = FALSE, ...)
```

Arguments

x	MPSE or tbl_mpse object
taxa.class	character the name of taxonomy class level what you want to extract
topn	integer the number of the top most abundant, default is NULL.
rmun	logical whether to remove the unknown taxa, such as "g__un_xxx", default is FALSE (the unknown taxa class will be considered as 'Others').
...	additional parameters

Author(s)

Shuangbin Xu

---

mp_extract_assays *extract the abundance matrix from MPSE object or tbl_mpse object*

Description

extract the abundance matrix from MPSE object or tbl_mpse object

Usage

```
mp_extract_assays(x, .abundance, byRow = TRUE, ...)

## S4 method for signature 'MPSE'
mp_extract_assays(x, .abundance, byRow = TRUE, ...)

## S4 method for signature 'tbl_mpse'
mp_extract_assays(x, .abundance, byRow = TRUE, ...)
```

```
## S4 method for signature 'grouped_df_mpse'
mp_extract_assays(x, .abundance, byRow = TRUE, ...)
```

Arguments

x	MPSE or tbl_mpse object
.abundance	the name of abundance to be extracted.
byRow	logical if it is set TRUE, 'otu X sample' shape will return, else 'sample X otu' will return.
...	additional parameters.

Value

otu abundance a data.frame object

`mp_extract_dist` *extract the dist object from MPSE or tbl_mpse object*

Description

extract the dist object from MPSE or tbl_mpse object

Usage

```
mp_extract_dist(x, distmethod, type = "sample", .group = NULL, ...)

## S4 method for signature 'MPSE'
mp_extract_dist(x, distmethod, type = "sample", .group = NULL, ...)

## S4 method for signature 'tbl_mpse'
mp_extract_dist(x, distmethod, type = "sample", .group = NULL, ...)

## S4 method for signature 'grouped_df_mpse'
mp_extract_dist(x, distmethod, type = "sample", .group = NULL, ...)
```

Arguments

x	MPSE object or tbl_mpse object
distmethod	character the method of calculated distance.
type	character, which type distance to be extracted, 'sample' represents the distance between the samples based on feature abundance matrix, 'feature' represents the distance between the features based on feature abundance matrix, 'env' represents the the distance between the samples based on continuous environment factors, default is 'sample'.

.group	the column name of sample information, which only work with type='sample' or type='env', default is NULL, when it is provided, a tibble that can be visualized via ggplot2 will return.
...	additional parameters

Value

dist object or tbl_df object when .group is provided.

`mp_extract_feature` *extract the feature (OTU) information in MPSE object*

Description

extract the feature (OTU) information in MPSE object

Usage

```
mp_extract_feature(x, addtaxa = FALSE, ...)

## S4 method for signature 'MPSE'
mp_extract_feature(x, addtaxa = FALSE, ...)

## S4 method for signature 'tbl_mpse'
mp_extract_feature(x, addtaxa = FALSE, ...)

## S4 method for signature 'grouped_df_mpse'
mp_extract_feature(x, addtaxa = FALSE, ...)
```

Arguments

x	MPSE object
addtaxa	logical whether adding the taxonomy information default is FALSE.
...	additional arguments

Value

tbl_df contained feature (OTU) information.

mp_extract_internal_attr *Extracting the PCA, PCoA, etc results from MPSE or tbl_mpse object*

Description

Extracting the PCA, PCoA, etc results from MPSE or tbl_mpse object

Usage

```
mp_extract_internal_attr(x, name, ...)

## S4 method for signature 'MPSE'
mp_extract_internal_attr(x, name, ...)

## S4 method for signature 'tbl_mpse'
mp_extract_internal_attr(x, name, ...)

## S4 method for signature 'grouped_df_mpse'
mp_extract_internal_attr(x, name, ...)
```

Arguments

x	MPSE or tbl_mpse object
name	character 'PCA' or 'PCoA'
...	additional parameters

Value

prcomp or pcoa etc object

mp_extract_rarecurve *Extract the result of mp_cal_rarecurve with action="add" from MPSE or tbl_mpse object*

Description

Extract the result of mp_cal_rarecurve with action="add" from MPSE or tbl_mpse object

Usage

```
mp_extract_rarecurve(x, .rarecurve, ...)

## S4 method for signature 'MPSE'
mp_extract_rarecurve(x, .rarecurve, ...)

## S4 method for signature 'tbl_mpse'
mp_extract_rarecurve(x, .rarecurve, ...)

## S4 method for signature 'grouped_df_mpse'
mp_extract_rarecurve(x, .rarecurve, ...)
```

Arguments

x	MPSE object or tbl_mpse object
.rarecurve	the column name of rarecurve after run mp_cal_rarecurve with action="add".
...	additional parameter

Value

rarecurve object that be be visualized by ggrarecurve

---

mp_extract_refseq *Extract the representative sequences from MPSE object*

Description

Extract the representative sequences from MPSE object

Usage

```
mp_extract_refseq(x, ...)

## S4 method for signature 'MPSE'
mp_extract_refseq(x, ...)

## S4 method for signature 'tbl_mpse'
mp_extract_refseq(x, ...)

## S4 method for signature 'grouped_df_mpse'
mp_extract_refseq(x, ...)
```

Arguments

x	MPSE object
...	additional parameters, meaningless now.

mp_extract_sample *extract the sample information in MPSE object*

Description

extract the sample information in MPSE object

Usage

```
mp_extract_sample(x, ...)

## S4 method for signature 'MPSE'
mp_extract_sample(x, ...)

## S4 method for signature 'tbl_mpse'
mp_extract_sample(x, ...)

## S4 method for signature 'grouped_df_mpse'
mp_extract_sample(x, ...)
```

Arguments

x	MPSE object
...	additional arguments

Value

tbl_df contained sample information.

mp_extract_tree *extract the taxonomy tree in MPSE object*

Description

extract the taxonomy tree in MPSE object

Usage

```
mp_extract_tree(x, type = "taxatree", tip.level = "OTU", ...)

## S4 method for signature 'MPSE'
mp_extract_tree(x, type = "taxatree", tip.level = "OTU", ...)

## S4 method for signature 'tbl_mpse'
mp_extract_tree(x, type = "taxatree", tip.level = "OTU", ...)
```

```
## S4 method for signature 'grouped_df_mpse'
mp_extract_tree(x, type = "taxatree", tip.level = "OTU", ...)

mp_extract_taxatree(x, tip.level = "OTU", ...)

mp_extract_otutree(x, ...)
```

Arguments

x	MPSE object
type	character taxatree or otutree
tip.level	character This argument will keep the nodes belong to the tip.level as tip nodes when type is taxatree, default is OTU, which will return the taxa tree with OTU level as tips.
...	additional arguments

Value

taxatree treedata object

---

mp_filter_taxa *Filter OTU (Features) By Abundance Level*

Description

Filter OTU (Features) By Abundance Level

Usage

```
mp_filter_taxa(
  .data,
  .abundance = NULL,
  min.abun = 0,
  min.prop = 0.05,
  include.lowest = FALSE,
  ...
)

## S4 method for signature 'MPSE'
mp_filter_taxa(
  .data,
  .abundance = NULL,
  min.abun = 0,
  min.prop = 0.05,
  include.lowest = FALSE,
```

```
  ...
)

## S4 method for signature 'tbl_mpse'
mp_filter_taxa(
  .data,
  .abundance = NULL,
  min.abun = 0,
  min.prop = 0.05,
  include.lowest = FALSE,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_filter_taxa(
  .data,
  .abundance = NULL,
  min.abun = 0,
  min.prop = 0.05,
  include.lowest = FALSE,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse or grouped_df_mpse object.
.abundance	the column names of abundance, default is NULL, meaning the 'Abundance' column.
min.abun	numeric minimum abundance required for each one sample default is 0 (.abundance=Abundance or NULL), meaning the abundance of OTU (Features) for each one sample should be >= 0.
min.prop	numeric minimum proportion of samples that contains the OTU (Features) when min.prop larger than 1, meaning the minimum number of samples that contains the OTU (Features).
include.lowest	logical whether include the lower boundary of min.abun default is FALSE (> min.abun), if it is TRUE, meaning (>= min.abun).
...	additional parameters, meaningless now.

Author(s)

Shuangbin Xu

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %>% mp_filter_taxa(.abundance=Abundance, min.abun=1, min.prop=1)
# For tbl_mpse object.
mouse.time.mpse %>% as_tibble %>% mp_filter_taxa(.abundance=Abundance, min.abun=1, min.prop=1)
```

```
# This also can be done using group_by, filter of dplyr.
mouse.time.mpse %>%
 dplyr::group_by(OTU) %>%
 dplyr::filter(sum(Abundance>=1)>=1)
```

`mp_fortify` *mp_fortify*

Description

Fortify a model with data in MicrobiotaProcess

Usage

```
mp_fortify(model, ...)
```

Arguments

`model`	object
`...`	additional parameters

Value

data frame or tbl_df object

`mp_import_biom` *building MPSE object from biom-format file.*

Description

building MPSE object from biom-format file.

Usage

```
mp_import_biom(
  biomfilename,
  mapfilename = NULL,
  otutree = NULL,
  refseq = NULL,
  ...
)
```

Arguments

biomfilename	character the biom-format file path.
mapfilename	character, the file contained sample information, the tsv format, default is NULL.
otutree	treedata, phylo or character, the file contained reference sequences, or treedata object, which is the result parsed by functions of treeio, default is NULL.
refseq	XStringSet or character, the file contained the representation sequence file or XStringSet class to store the representation sequence, default is NULL.
...	additional parameter, which is meaningless now.

Value

MPSE-class

---

mp_import_humann_regroup

Import function to load the output of human_regroup_table in HUMAnN.

Description

Import function to load the output of human_regroup_table in HUMAnN.

Usage

```
mp_import_humann_regroup(
  profile,
  mapfilename = NULL,
  rm.unknown = TRUE,
  keep.contribute.abundance = FALSE,
  ...
)
```

Arguments

profile	the output file (text format) of human_regroup_table in HUMAnN.
mapfilename	the sample information file or data.frame,
rm.unknown	logical whether remove the unmapped and ungrouped features.
keep.contribute.abundance	logical whether keep the abundance of contributed taxa, default is FALSE, it will consume more memory if it set to TRUE.
...	additional parameters, meaningless now.

Author(s)

Shuangbin Xu

`mp_import_metaphlan` *Import function to load the output of MetaPhlAn.*

Description

Import function to load the output of MetaPhlAn.

Usage

```
mp_import_metaphlan(
  profile,
  mapfilename = NULL,
  treefile = NULL,
  linenum = NULL,
  ...
)
```

Arguments

`profile`	the output file (text format) of MetaPhlAn.
`mapfilename`	the sample information file or data.frame, default is NULL.
`treefile`	the path of MetaPhlAn tree file (mpa_v30_CHOCOPhlAn_201901_species_tree.nwk), default is NULL.

`linenum`	a integer, sometimes the output file of MetaPhlAn (< 3) contained the sample information in the first several lines. The linenum should be required. for example:
	`group A A A A B B B B`
	`subgroup A1 A1 A2 A2 B1 B1 B2 B2`
	`subject S1 S2 S3 S4 S5 S6 S7 S8`
	`Bacteria 99 99 99 99 99 99 99 99`
	`...`
	the `linenum` should be set to 3.
	`sampleid A1 A2 A3 A4 A5`
	`Bacteria 99 99 99 99 99`
	`...`
	The `linenum` should be set to 1.

`...`	additional parameters, meaningless now.

Details

When the output abundance of MetaPhlAn is relative abundance, the `force` of `mp_cal_abundance` should be set to TRUE, and the `relative` of `mp_cal_abundance` should be set to FALSE. Because the abundance profile will be rarefied in the default (force=FALSE), which requires the integer (count) abundance, then the relative abundance will be calculated in the default (relative=TRUE).

Author(s)

Shuangbin Xu

Examples

```
file1 <- system.file("extdata/MetaPhlAn", "metaphlan_test.txt", package="MicrobiotaProcess")
sample.file <- system.file("extdata/MetaPhlAn", "sample_test.txt", package="MicrobiotaProcess")
readLines(file1, n=3) %>% writeLines()
mpse1 <- mp_import_metaphlan(profile=file1, mapfilename=sample.file)
mpse1
```

---

mp_import_qiime *Import function to load the output of qiime.*

Description

The function was designed to import the output of qiime and convert them to MPSE class.

Usage

```
mp_import_qiime(
  otufilename,
  mapfilename = NULL,
  otutree = NULL,
  refseq = NULL,
  ...
)
```

Arguments

otufilename	character, the file contained otu table, the ouput of qiime.
mapfilename	character, the file contained sample information, the tsv format, default is NULL.
otutree	treedata, phylo or character, the file contained reference sequences, or treedata object, which is the result parsed by functions of treeio, default is NULL.
refseq	XStringSet or character, the file contained the representation sequence file or XStringSet class to store the representation sequence, default is NULL.
...	additional parameters.

Value

MPSE-class.

Author(s)

Shuangbin Xu

mp_mantel *Mantel and Partial Mantel Tests for MPSE or tbl_mpse Object*

Description

Mantel and Partial Mantel Tests for MPSE or tbl_mpse Object

Usage

```
mp_mantel(
  .data,
  .abundance,
  .y.env,
  .z.env = NULL,
  distmethod = "bray",
  distmethod.y = "euclidean",
  distmethod.z = "euclidean",
  method = "pearson",
  permutations = 999,
  action = "get",
  seed = 123,
  scale.y = FALSE,
  scale.z = FALSE,
  ...
)

## S4 method for signature 'MPSE'
mp_mantel(
  .data,
  .abundance,
  .y.env,
  .z.env = NULL,
  distmethod = "bray",
  distmethod.y = "euclidean",
  distmethod.z = "euclidean",
  method = "pearson",
  permutations = 999,
  action = "get",
  seed = 123,
  scale.y = FALSE,
  scale.z = FALSE,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_mantel(
  .data,
```

```
  .abundance,
  .y.env,
  .z.env = NULL,
  distmethod = "bray",
  distmethod.y = "euclidean",
  distmethod.z = "euclidean",
  method = "pearson",
  permutations = 999,
  action = "get",
  seed = 123,
  scale.y = FALSE,
  scale.z = FALSE,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_mantel(
  .data,
  .abundance,
  .y.env,
  .z.env = NULL,
  distmethod = "bray",
  distmethod.y = "euclidean",
  distmethod.z = "euclidean",
  method = "pearson",
  permutations = 999,
  action = "get",
  seed = 123,
  scale.y = FALSE,
  scale.z = FALSE,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of otu abundance to be calculated
.y.env	the column names of continuous environment factors to perform Mantel statistic, it is required.
.z.env	the column names of continuous environment factors to perform Partial Mantel statistic based on this, default is NULL.
distmethod	character the method to calculate distance based on .abundance.
distmethod.y	character the method to calculate distance based on .y.env.
distmethod.z	character the method of calculated distance based on .z.env
method	character Correlation method, options is "pearson", "spearman" or "kendall"
permutations	the number of permutations required, default is 999.

action	character, "add" joins the mantel result to the internal attributes of the object, "only" and "get" return 'mantel' or 'mantel.partial' (if .z.env is provided) object.
seed	a random seed to make the analysis reproducible, default is 123.
scale.y	logical whether scale the environment matrix (.y.env) before the distance is calculated, default is FALSE
scale.z	logical whether scale the environment matrix (.z.env) before the distance is calculated, default is FALSE
...	additional parameters, see also mantel.

Value

update object or tibble according the 'action'

See Also

mantel

Examples

```
library(vegan)
data(varespec, varechem)
mpse <- MPSE(assays=list(Abundance=t(varespec)), colData=varechem)
mpse %>% mp_mantel(.abundance=Abundance,
                   .y.env=colnames(varechem),
                   distmethod.y="euclidean",
                   scale.y = TRUE
                   )
```

---

mp_mrpp *Analysis of Multi Response Permutation Procedure (MRPP) with MPSE or tbl_mpse object*

Description

Analysis of Multi Response Permutation Procedure (MRPP) with MPSE or tbl_mpse object

Usage

```
mp_mrpp(
  .data,
  .abundance,
  .group,
  distmethod = "bray",
  action = "add",
  permutations = 999,
  seed = 123,
  ...
```

```
)

## S4 method for signature 'MPSE'
mp_mrpp(
  .data,
  .abundance,
  .group,
  distmethod = "bray",
  action = "add",
  permutations = 999,
  seed = 123,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_mrpp(
  .data,
  .abundance,
  .group,
  distmethod = "bray",
  action = "add",
  permutations = 999,
  seed = 123,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_mrpp(
  .data,
  .abundance,
  .group,
  distmethod = "bray",
  action = "add",
  permutations = 999,
  seed = 123,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of abundance to be calculated.
.group	The name of the column of the sample group information.
distmethod	character the method to calculate pairwise distances, default is 'bray'.
action	character "add" joins the ANOSIM result to internal attribute of the object, "only" return a tibble contained the statistic information of MRPP analysis, and "get" return 'mrpp' object can be analyzed using the related vegan funtion.

permutations	the number of permutations required, default is 999.
seed	a random seed to make the MRPP analysis reproducible, default is 123.
...	additional parameters see also 'mrpp' of vegan.

Value

update object according action argument

Author(s)

Shuangbin

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %>%
  mp_decostand(.abundance=Abundance) %>%
  mp_mrpp(.abundance=hellinger,
          .group=time,
          distmethod="bray",
          permutations=999, # for more robust, set it to 9999.
          action="get")
```

---

mp_plot_abundance *plotting the abundance of taxa via specified taxonomy class*

Description

plotting the abundance of taxa via specified taxonomy class

Usage

```
mp_plot_abundance(
  .data,
  .abundance = NULL,
  .group = NULL,
  taxa.class = NULL,
  topn = 10,
  relative = TRUE,
  force = FALSE,
  plot.group = FALSE,
  geom = "flowbar",
  feature.dist = "bray",
  feature.hclust = "average",
  sample.dist = "bray",
  sample.hclust = "average",
  .sec.group = NULL,
  rmun = FALSE,
```

```
  rm.zero = TRUE,
  order.by.feature = FALSE,
  ...
)

## S4 method for signature 'MPSE'
mp_plot_abundance(
  .data,
  .abundance = NULL,
  .group = NULL,
  taxa.class = NULL,
  topn = 10,
  relative = TRUE,
  force = FALSE,
  plot.group = FALSE,
  geom = "flowbar",
  feature.dist = "bray",
  feature.hclust = "average",
  sample.dist = "bray",
  sample.hclust = "average",
  .sec.group = NULL,
  rmun = FALSE,
  rm.zero = TRUE,
  order.by.feature = FALSE,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_plot_abundance(
  .data,
  .abundance = NULL,
  .group = NULL,
  taxa.class = NULL,
  topn = 10,
  relative = TRUE,
  force = FALSE,
  plot.group = FALSE,
  geom = "flowbar",
  feature.dist = "bray",
  feature.hclust = "average",
  sample.dist = "bray",
  sample.hclust = "average",
  .sec.group = NULL,
  rmun = FALSE,
  rm.zero = TRUE,
  order.by.feature = FALSE,
  ...
)
```

```
## S4 method for signature 'grouped_df_mpse'
mp_plot_abundance(
  .data,
  .abundance = NULL,
  .group = NULL,
  taxa.class = NULL,
  topn = 10,
  relative = TRUE,
  force = FALSE,
  plot.group = FALSE,
  geom = "flowbar",
  feature.dist = "bray",
  feature.hclust = "average",
  sample.dist = "bray",
  sample.hclust = "average",
  .sec.group = NULL,
  rmun = FALSE,
  rm.zero = TRUE,
  order.by.feature = FALSE,
  ...
)
```

Arguments

.data	MPSE object or tbl_mpse object
.abundance	the column name of abundance to be plotted.
.group	the column name of group to be calculated and plotted, default is NULL.
taxa.class	name of taxonomy class, default is NULL, meaning the Phylum class will be plotted.
topn	integer the number of the top most abundant, default is 10.
relative	logical whether calculate the relative abundance and plotted.
force	logical whether calculate the relative abundance forcibly when the abundance is not be rarefied, default is FALSE.
plot.group	logical whether plotting the abundance of specified taxa.class taxonomy with group not sample level, default is FALSE.
geom	character which type plot, options is 'flowbar' 'bar' and 'heatmap', default is 'flowbar'.
feature.dist	character the method to calculate the distance between the features, based on the '.abundance' of 'taxa.class', default is 'bray', options refer to the 'distmethod' of [mp_cal_dist()] (except unifrac related).
feature.hclust	character the agglomeration method for the features, default is 'average', options are 'single', 'complete', 'average', 'ward.D', 'ward.D2', 'centroid' 'median' and 'mcquitty'.

`sample.dist` character the method to calculate the distance between the samples based on the '.abundance' of 'taxa.class', default is 'bray', options refer to the 'distmethod' of [mp_cal_dist()] (except unifrac related).

`sample.hclust` character the agglomeration method for the samples, default is 'average', options are 'single', 'complete', 'average', 'ward.D', 'ward.D2', 'centroid' 'median' and 'mcquitty'.

`.sec.group` the column name of second group to be plotted with nested facet, default is NULL, this argument will be deprecated in the next version.

`rmun` logical whether to group the unknown taxa to `Others` category, such as "g__un_xxx", default is FALSE, meaning do not group them to `Others` category.

`rm.zero` logical whether to display the zero abundance, which only work with geom='heatmap' default is TRUE.

`order.by.feature` character adjust the order of axis x, default is FALSE, if it is NULL or TRUE, meaning the order of axis.x will be visualizing with the order of samples by highest abundance of features.

`...` additional parameters, when the geom = "flowbar", it can specify the parameters of 'geom_stratum' of 'ggalluvial', when the geom = 'bar', it can specify the parameters of 'geom_bar' of 'ggplot2', when the geom = "heatmap", it can specify the parameter of 'geom_tile' of 'ggplot2'.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(mouse.time.mpse)
mouse.time.mpse %<>%
  mp_rrarefy()
mouse.time.mpse
mouse.time.mpse %<>%
  mp_cal_abundance(.abundance=RareAbundance, action="add") %>%
  mp_cal_abundance(.abundance=RareAbundance, .group=time, action="add")
mouse.time.mpse
p1 <- mouse.time.mpse %>%
      mp_plot_abundance(.abundance=RelRareAbundanceBySample,
                        .group=time,
                        taxa.class="Phylum",
                        topn=20)
p2 <- mouse.time.mpse %>%
      mp_plot_abundance(.abundance = Abundance,
                        taxa.class = Phylum,
                        topn = 20,
                        relative = FALSE,
                        force = TRUE
                       )
p3 <- mouse.time.mpse %>%
```

```
      mp_plot_abundance(.abundance = RareAbundance,
                        .group = time,
                        taxa.class = Phylum,
                        topn = 20,
                        relative = FALSE,
                        force = TRUE
                        )
p4 <- mouse.time.mpse %>%
      mp_plot_abundance(.abundance = RareAbundance,
                        .group = time,
                        taxa.class = Phylum,
                        topn = 20,
                        relative = FALSE,
                        force = TRUE,
                        plot.group = TRUE
                        )

## End(Not run)
```

---

mp_plot_alpha *Plotting the alpha diversity between samples or groups.*

Description

Plotting the alpha diversity between samples or groups.

Usage

```
mp_plot_alpha(
  .data,
  .group,
  .alpha = c("Observe", "Shannon"),
  test = "wilcox.test",
  comparisons = NULL,
  step_increase = 0.05,
  ...
)

## S4 method for signature 'MPSE'
mp_plot_alpha(
  .data,
  .group,
  .alpha = c("Observe", "Shannon"),
  test = "wilcox.test",
  comparisons = NULL,
  step_increase = 0.05,
  ...
)
```

```
## S4 method for signature 'tbl_mpse'
mp_plot_alpha(
  .data,
  .group,
  .alpha = c("Observe", "Shannon"),
  test = "wilcox.test",
  comparisons = NULL,
  step_increase = 0.05,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_plot_alpha(
  .data,
  .group,
  .alpha = c("Observe", "Shannon"),
  test = "wilcox.test",
  comparisons = NULL,
  step_increase = 0.05,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.group	the column name of sample group information
.alpha	the column name of alpha index after run mp_cal_alpha or mp_cal_pd_metric.
test	the name of the statistical test, default is 'wilcox.test'
comparisons	A list of length-2 vectors. The entries in the vector are either the names of 2 values on the x-axis or the 2 integers that correspond to the index of the columns of interest, default is NULL, meaning it will be calculated automatically with the names in the .group.
step_increase	numeric vector with the increase in fraction of total height for every additional comparison to minimize overlap, default is 0.05.
...	additional parameters, see also geom_signif

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(mouse.time.mpse)
mpse <- mouse.time.mpse %>%
        mp_rrarefy() %>%
        mp_cal_alpha(.abundance=RareAbundance)
```

```
mpse
p <- mpse %>%
     mp_plot_alpha(.group=time, .alpha=c(Observe, Shannon, Pielou))
p

## End(Not run)
```

`mp_plot_diff_boxplot` *displaying the differential result contained abundance and LDA with boxplot (abundance) and error bar (LDA).*

Description

displaying the differential result contained abundance and LDA with boxplot (abundance) and error bar (LDA).

Usage

```
mp_plot_diff_boxplot(
  .data,
  .group,
  .size = 2,
  errorbar.xmin = NULL,
  errorbar.xmax = NULL,
  point.x = NULL,
  taxa.class = "all",
  group.abun = FALSE,
  removeUnknown = FALSE,
  ...
)

## S4 method for signature 'MPSE'
mp_plot_diff_boxplot(
  .data,
  .group,
  .size = 2,
  errorbar.xmin = NULL,
  errorbar.xmax = NULL,
  point.x = NULL,
  taxa.class = "all",
  group.abun = FALSE,
  removeUnknown = FALSE,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_plot_diff_boxplot(
```

```
  .data,
  .group,
  .size = 2,
  errorbar.xmin = NULL,
  errorbar.xmax = NULL,
  point.x = NULL,
  taxa.class = "all",
  group.abun = FALSE,
  removeUnknown = FALSE,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_plot_diff_boxplot(
  .data,
  .group,
  .size = 2,
  errorbar.xmin = NULL,
  errorbar.xmax = NULL,
  point.x = NULL,
  taxa.class = "all",
  group.abun = FALSE,
  removeUnknown = FALSE,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse after run mp_diff_analysis with 'action="add"'.
.group	the column name for mapping the different color.
.size	the column name for mapping the size of points or numeric, default is 2.
errorbar.xmin	the column name for 'xmin' mapping of error barplot layer, default is NULL.
errorbar.xmax	the column name for 'xmax' mapping of error barplot layer, default is NULL.
point.x	the column name for 'x' mapping of point layer (right panel), default is NULL.
taxa.class	the taxonomy class features will be displayed, default is 'all'.
group.abun	logical whether plot the abundance in each group with bar plot, default is FALSE.
removeUnknown	logical whether mask the unknown taxonomy information but differential species, default is FALSE.
...	additional params, see also the 'geom_boxplot', 'geom_errorbarh' and 'geom_point'.

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %<>%
  mp_rrarefy()
mouse.time.mpse
mouse.time.mpse %<>%
```

```
  mp_diff_analysis(.abundance=RareAbundance,
                   .group=time,
                   first.test.alpha=0.01,
                   action="add")
library(ggplot2)
p1 <- mouse.time.mpse %>%
        mp_plot_diff_boxplot(.group = time) %>%
        set_diff_boxplot_color(
          values = c("deepskyblue", "orange"),
          guide = guide_legend(title=NULL)
        )
p1
p2 <- mouse.time.mpse %>%
        mp_plot_diff_boxplot(
          taxa.class = c(Genus, OTU),
          group.abun = TRUE,
          removeUnknown = TRUE,
        ) %>%
        set_diff_boxplot_color(
          values = c("deepskyblue", "orange"),
          guide = guide_legend(title=NULL)
        )
p2
```

`mp_plot_diff_cladogram` *Visualizing the result of mp_diff_analysis with cladogram.*

Description

Visualizing the result of mp_diff_analysis with cladogram.

Usage

```
mp_plot_diff_cladogram(
  .data,
  .group,
  .size = "pvalue",
  taxa.class,
  removeUnknown = FALSE,
  layout = "radial",
  hilight.alpha = 0.3,
  hilight.size = 0.2,
  bg.tree.size = 0.15,
  bg.tree.color = "#bed0d1",
  bg.point.color = "#bed0d1",
  bg.point.fill = "white",
  bg.point.stroke = 0.2,
  bg.point.size = 2,
```

```
  label.size = 2.6,
  tip.annot = TRUE,
  as.tiplab = TRUE,
  ...
)
```

Arguments

.data	MPSE object or treedata which was from the taxatree slot after running the 'mp_diff_analysis'.
.group	the column name for mapping the different color.
.size	the column name for mapping the size of points, default is 'pvalue'.
taxa.class	the taxonomy class name will be replaced shorthand, default is the one level above 'OTU'.
removeUnknown	logical, whether mask the unknown taxonomy information but differential species, default is FALSE.
layout	character, the layout of tree, default is 'radial', see also the 'layout' of 'ggtree'.
hilight.alpha	numeric, the transparency of high light clade, default is 0.3.
hilight.size	numeric, the margin thickness of high light clade, default is 0.2.
bg.tree.size	numeric, the line size (width) of tree, default is 0.15.
bg.tree.color	character, the line color of tree, default is '#bed0d1'.
bg.point.color	character, the color of margin of background node points of tree, default is '#bed0d1'.
bg.point.fill	character, the point fill (since point shape is 21) of background nodes of tree, default is 'white'.
bg.point.stroke	numeric, the margin thickness of point of background nodes of tree, default is 0.2 .
bg.point.size	numeric, the point size of background nodes of tree, default is 2.
label.size	numeric, the label size of differential taxa, default is 2.6.
tip.annot	logcial whether to replace the differential tip labels with shorthand, default is TRUE.
as.tiplab	logical, whether to display the differential tip labels with 'geom_tiplab' of 'ggtree', default is TRUE, if it is FALSE, it will use 'geom_text_repel' of 'ggrepel'.
...	additional parameters, meaningless now.

Details

The color scale of differential group can be designed by 'scale_fill_diff_cladogram'

Examples

```
## Not run:
  data(mouse.time.mpse)
  mouse.time.mpse %<>%
    mp_rrarefy()
  mouse.time.mpse
  mouse.time.mpse %<>%
    mp_diff_analysis(.abundance=RareAbundance,
                     .group=time,
                     first.test.alpha=0.01,
                     action="add")
  #' ### visualizing the differential taxa with cladogram
  library(ggplot2)
  f <- mouse.time.mpse %>%
       mp_plot_diff_cladogram(
         label.size = 2.5,
         hilight.alpha = .3,
         bg.tree.size = .5,
         bg.point.size = 2,
         bg.point.stroke = .25
       ) +
       scale_fill_diff_cladogram(
         values = c('skyblue', 'orange')
       ) +
       scale_size_continuous(range = c(1, 4))
  f

## End(Not run)
```

---

mp_plot_diff_manhattan

displaying the differential result contained abundance and LDA with manhattan plot.

Description

displaying the differential result contained abundance and LDA with manhattan plot.

Usage

```
mp_plot_diff_manhattan(
  .data,
  .group,
  .y = "fdr",
  .size = 2,
  taxa.class = "OTU",
  anno.taxa.class = NULL,
  removeUnknown = FALSE,
  ...
```

```
)

## S4 method for signature 'MPSE'
mp_plot_diff_manhattan(
  .data,
  .group,
  .y = "fdr",
  .size = 2,
  taxa.class = "OTU",
  anno.taxa.class = NULL,
  removeUnknown = FALSE,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_plot_diff_manhattan(
  .data,
  .group,
  .y = "fdr",
  .size = 2,
  taxa.class = "OTU",
  anno.taxa.class = NULL,
  removeUnknown = FALSE,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_plot_diff_manhattan(
  .data,
  .group,
  .y = "fdr",
  .size = 2,
  taxa.class = "OTU",
  anno.taxa.class = NULL,
  removeUnknown = FALSE,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse after run 'mp_diff_analysis' with 'action="add"'.
.group	the column name for mapping the different color.
.y	the column name for mapping the y axis, default is 'fdr'.
.size	the column name for mapping the size of points or numeric, default is 2.
taxa.class	the taxonomy class features will be displayed, default is 'OTU'.
anno.taxa.class	the taxonomy class to annotate the sign taxa with color, default is 'Phylum' if 'taxatree' is not empty.

removeUnknown	logical whether mask the unknown taxonomy information but differential species, default is FALSE.
...	additional params, see also the 'geom_text_repel' and 'geom_point'.

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %<>%
  mp_rrarefy()
mouse.time.mpse
mouse.time.mpse %<>%
  mp_diff_analysis(.abundance=RareAbundance,
                   .group=time,
                   first.test.alpha=0.01,
                   action="add")
p <- mouse.time.mpse %>%
       mp_plot_diff_manhattan(
           .group = Sign_time,
           .y = fdr,
           .size = 2,
           taxa.class = OTU,
           anno.taxa.class = Phylum,
       )
```

---

mp_plot_diff_res *The visualization of result of mp_diff_analysis*

Description

The visualization of result of mp_diff_analysis

Usage

```
mp_plot_diff_res(
  .data,
  .group,
  layout = "radial",
  tree.type = "taxatree",
  .taxa.class = NULL,
  barplot.x = NULL,
  point.size = NULL,
  sample.num = 50,
  tiplab.size = 2,
  offset.abun = 0.04,
  pwidth.abun = 0.8,
  offset.effsize = 0.3,
  pwidth.effsize = 0.5,
  group.abun = FALSE,
```

```
  tiplab.linetype = 3,
  ...
)

## S4 method for signature 'MPSE'
mp_plot_diff_res(
  .data,
  .group,
  layout = "radial",
  tree.type = "taxatree",
  .taxa.class = NULL,
  barplot.x = NULL,
  point.size = NULL,
  sample.num = 50,
  tiplab.size = 2,
  offset.abun = 0.04,
  pwidth.abun = 0.8,
  offset.effsize = 0.3,
  pwidth.effsize = 0.5,
  group.abun = FALSE,
  tiplab.linetype = 3,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_plot_diff_res(
  .data,
  .group,
  layout = "radial",
  tree.type = "taxatree",
  .taxa.class = NULL,
  barplot.x = NULL,
  point.size = NULL,
  sample.num = 50,
  tiplab.size = 2,
  offset.abun = 0.04,
  pwidth.abun = 0.8,
  offset.effsize = 0.3,
  pwidth.effsize = 0.5,
  group.abun = FALSE,
  tiplab.linetype = 3,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_plot_diff_res(
  .data,
  .group,
```

```
  layout = "radial",
  tree.type = "taxatree",
  .taxa.class = NULL,
  barplot.x = NULL,
  point.size = NULL,
  sample.num = 50,
  tiplab.size = 2,
  offset.abun = 0.04,
  pwidth.abun = 0.8,
  offset.effsize = 0.3,
  pwidth.effsize = 0.5,
  group.abun = FALSE,
  tiplab.linetype = 3,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse after run mp_diff_analysis with action="add"
.group	the column name for mapping the different color, default is the column name has 'Sign_' prefix, which contains the enriched group name, but the insignificant should be NA.
layout	the type of tree layout, should be one of "rectangular", "roundrect", "ellipse", "circular", "slanted", "radial", "inward_circular".
tree.type	one of 'taxatree' and 'otutree', taxatree is the taxonomy class tree 'otutree' is the phylogenetic tree built with the representative sequences.
.taxa.class	character the name of taxonomy class level, default is NULL, meaning it will extract the phylum annotation automatically.
barplot.x	the column name of continuous value mapped to barplot, default is NULL, meaning the 'LDAmean' will be used internally.
point.size	the column name of continuous value mapped to the size of point in the tree, default is NULL, meaning the 'fdr' will be used internally.
sample.num	integer when it is smaller than the sample number of '.data', the abundance of '.group' will replace the abundance of sample, default is 50.
tiplab.size	numeric the size of tiplab, default is 2.
offset.abun	numeric the gap (width) (relative width to tree) between the tree and abundance panel, default is 0.04.
pwidth.abun	numeric the panel width (relative width to tree) of abundance panel, default is 0.3 .
offset.effsize	numeric the gap (width) (relative width to tree) between the tree and effect size panel, default is 0.3 .
pwidth.effsize	numeric the panel width (relative width to tree) of effect size panel, default is 0.5 .
group.abun	logical whether to display the relative abundance of group instead of sample, default is FALSE.

tiplab.linetype
: numeric the type of line for adding line if 'tree.type' is 'otutree', default is 3 .

... : additional parameters, meaningless now.

---

mp_plot_dist *Plotting the distance between the samples with heatmap or boxplot.*

Description

Plotting the distance between the samples with heatmap or boxplot.

Usage

```
mp_plot_dist(
  .data,
  .distmethod,
  .group = NULL,
  group.test = FALSE,
  hclustmethod = "average",
  test = "wilcox.test",
  comparisons = NULL,
  step_increase = 0.1,
  ...
)

## S4 method for signature 'MPSE'
mp_plot_dist(
  .data,
  .distmethod,
  .group = NULL,
  group.test = FALSE,
  hclustmethod = "average",
  test = "wilcox.test",
  comparisons = NULL,
  step_increase = 0.1,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_plot_dist(
  .data,
  .distmethod,
  .group = NULL,
  group.test = FALSE,
  hclustmethod = "average",
  test = "wilcox.test",
  comparisons = NULL,
```

```
  step_increase = 0.1,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_plot_dist(
  .data,
  .distmethod,
  .group = NULL,
  group.test = FALSE,
  hclustmethod = "average",
  test = "wilcox.test",
  comparisons = NULL,
  step_increase = 0.1,
  ...
)
```

Arguments

.data	the MPSE or tbl_mpse object after [mp_cal_dist()] is performed with action="add"
.distmethod	the column names of distance of samples, it will generate after [mp_cal_dist()] is performed.
.group	the column names of group, default is NULL, when it is not provided the heatmap of distance between samples will be returned. If it is provided and group.test is TURE, the comparisons boxplot of distance between the group will be returned, but when group.test is FALSE, the heatmap of distance between samples with group information will be returned.
group.test	logical default is FALSE, see the .group argument.
hclustmethod	character the method of hclust, default is 'average' (= UPGMA).
test	the name of the statistical test, default is 'wilcox.test'
comparisons	A list of length-2 vectors. The entries in the vector are either the names of 2 values on the x-axis or the 2 integers that correspond to the index of the columns of interest, default is NULL, meaning it will be calculated automatically with the names in the .group.
step_increase	numeric vector with the increase in fraction of total height for every additional comparison to minimize overlap, default is 0.1.
...	additional parameters, see also geom_signif

Author(s)

Shuangbin Xu

See Also

[mp_cal_dist()] and [mp_extract_dist()]

Examples

```
## Not run:
data(mouse.time.mpse)
mouse.time.mpse %<>% mp_decostand(.abundance=Abundance)
mouse.time.mpse
mouse.time.mpse %<>%
  mp_cal_dist(.abundance=hellinger, distmethod="bray")
mouse.time.mpse
p1 <- mouse.time.mpse %>%
        mp_plot_dist(.distmethod=bray)
p2 <- mouse.time.mpse %>%
        mp_plot_dist(.distmethod=bray, .group=time, group.test=TRUE)
p3 <- mouse.time.mpse %>%
        mp_plot_dist(.distmethod=bray, .group=time)

## End(Not run)
```

`mp_plot_ord` *Plotting the result of PCA, PCoA, CCA, RDA, NDMS or DCA*

Description

Plotting the result of PCA, PCoA, CCA, RDA, NDMS or DCA

Usage

```
mp_plot_ord(
  .data,
  .ord,
  .dim = c(1, 2),
  .group = NULL,
  .starshape = 15,
  .size = 2,
  .alpha = 1,
  .color = "black",
  starstroke = 0.5,
  show.side = TRUE,
  show.adonis = FALSE,
  ellipse = FALSE,
  show.sample = FALSE,
  show.envfit = FALSE,
  p.adjust = NULL,
  filter.envfit = FALSE,
  ...
)

## S4 method for signature 'MPSE'
mp_plot_ord(
```

```
  .data,
  .ord,
  .dim = c(1, 2),
  .group = NULL,
  .starshape = 15,
  .size = 2,
  .alpha = 1,
  .color = "black",
  starstroke = 0.5,
  show.side = TRUE,
  show.adonis = FALSE,
  ellipse = FALSE,
  show.sample = FALSE,
  show.envfit = FALSE,
  p.adjust = NULL,
  filter.envfit = FALSE,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_plot_ord(
  .data,
  .ord,
  .dim = c(1, 2),
  .group = NULL,
  .starshape = 15,
  .size = 2,
  .alpha = 1,
  .color = "black",
  starstroke = 0.5,
  show.side = TRUE,
  show.adonis = FALSE,
  ellipse = FALSE,
  show.sample = FALSE,
  show.envfit = FALSE,
  p.adjust = NULL,
  filter.envfit = FALSE,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_plot_ord(
  .data,
  .ord,
  .dim = c(1, 2),
  .group = NULL,
  .starshape = 15,
  .size = 2,
```

```
  .alpha = 1,
  .color = "black",
  starstroke = 0.5,
  show.side = TRUE,
  show.adonis = FALSE,
  ellipse = FALSE,
  show.sample = FALSE,
  show.envfit = FALSE,
  p.adjust = NULL,
  filter.envfit = FALSE,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object, it is required.
.ord	a name of ordination (required), options are PCA, PCoA, DCA, NMDS, RDA, CCA, but the corresponding calculation methods (mp_cal_pca, mp_cal_pcoa, ...) should be done with action="add" before it.
.dim	integer which dimensions will be displayed, it should be a vector (length=2) default is c(1, 2). if the length is one the default will also be displayed.
.group	the column name of variable to be mapped to the color of points (fill character of geom_star) or one specified color code, default is NULL, meaning fill=NA, the points are hollow.
.starshape	the column name of variable to be mapped to the shapes of points (starshape character of geom_star) or one specified starshape of point of ggstar, default is NULL, meaning starshape=15 (circle point).
.size	the column name of variable to be mapped to the size of points (size character of geom_star) or one specified size of point of ggstar, default is NULL, meaning the size=1.5, the size of points.
.alpha	the column name of variable to be mapped to the transparency of points (alpha character of geom_star) or one specified alpha of point of ggstar. default is NULL, meaning the alpha=1, the transparency of points.
.color	the column name of variable to be mapped to the color of line of points (color character of geom_star) or one specified starshape of point of ggstar, default is NULL, meaning the color is 'black'.
starstroke	numeric the width of edge of points, default is 0.5.
show.side	logical whether display the side boxplot with the specified .dim dimensions, default is TRUE.
show.adonis	logical whether display the result of mp_adonis with action='all', default is FALSE.
ellipse	logical, whether to plot ellipses, default is FALSE. (.group or .color variables according to the 'geom', the default geom is path, so .color can be mapped to the corresponding variable).
show.sample	logical, whether display the sample names of points, default is FALSE.

show.envfit	logical, whether display the result after run [mp_envfit()], default is FALSE.
p.adjust	a character method of p.adjust p.adjust, default is NULL, options are 'fdr', 'bonferroni', 'BH' etc.
filter.envfit	logical or numeric, whether to remove the no significant environment factor after run [mp_envfit()], default is FALSE, meaning do not remove. If it is numeric, meaning the keep p.value or the adjust p with p.adjust the factors smaller than the numeric, e.g when filter.envfit=0.05 or (filter.envfit=TRUE), meaning the factors of p <= 0.05 will be displayed.
...	additional parameters, see also the stat_ellipse.

See Also

[mp_cal_pca()], [mp_cal_pcoa], [mp_cal_nmds], [mp_cal_rda], [mp_cal_cca], [mp_envfit()] and [mp_extract_internal_attr()]

Examples

```
## Not run:
library(vegan)
data(varespec, varechem)
mpse <- MPSE(assays=list(Abundance=t(varespec)), colData=varechem)
envformula <- paste("~", paste(colnames(varechem), collapse="+")) %>% as.formula
mpse %<>%
mp_cal_cca(.abundance=Abundance, .formula=envformula, action="add") %>%
mp_envfit(.ord=CCA, .env=colnames(varechem), permutations=9999, action="add")
mpse
p1 <- mpse %>% mp_plot_ord(.ord=CCA, .group=Al, .size=Mn)
p1
p2 <- mpse %>% mp_plot_ord(.ord=CCA, .group=Al, .size=Mn, show.sample=TRUE)
p2
p3 <- mpse %>% mp_plot_ord(.ord=CCA, .group="blue", .size=Mn, .alpha=0.8, show.sample=TRUE)
p3
p4 <- mpse %>% mp_plot_ord(.ord=CCA, .group=Al, .size=Mn, show.sample=TRUE, show.envfit=TRUE)
p4

## End(Not run)
```

---

mp_plot_rarecurve *Rarefaction alpha index with MPSE*

Description

Rarefaction alpha index with MPSE

Usage

```
mp_plot_rarecurve(
  .data,
  .rare,
  .alpha = c("Observe", "Chao1", "ACE"),
  .group = NULL,
  nrow = 1,
  plot.group = FALSE,
  ...
)

## S4 method for signature 'MPSE'
mp_plot_rarecurve(
  .data,
  .rare,
  .alpha = c("Observe", "Chao1", "ACE"),
  .group = NULL,
  nrow = 1,
  plot.group = FALSE,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_plot_rarecurve(
  .data,
  .rare,
  .alpha = c("Observe", "Chao1", "ACE"),
  .group = NULL,
  nrow = 1,
  plot.group = FALSE,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_plot_rarecurve(
  .data,
  .rare,
  .alpha = c("Observe", "Chao1", "ACE"),
  .group = NULL,
  nrow = 1,
  plot.group = FALSE,
  ...
)
```

Arguments

.data	MPSE object or tbl_mpse after it was performed mp_cal_rarecurve with action='add'
.rare	the column names of

.alpha	the names of alpha index, which should be one or more of Observe, ACE, Chao1, default is Observe.
.group	the column names of group, default is NULL, when it is provided, the rarecurve lines will group and color with the group.
nrow	integer Number of rows in facet_wrap.
plot.group	logical whether to combine the samples, default is FALSE, when it is TRUE, the samples of same group will be represented by their group.
...	additional parameters, see also geom_smooth.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(mouse.time.mpse)
mpse <- mouse.time.mpse %>%
        mp_rrarefy()
mpse
mpse %<>% mp_cal_rarecurve(.abundance=RareAbundance, chunks=100, action="add")
mpse
p1 <- mpse %>% mp_plot_rarecurve(.rare=RareAbundanceRarecurve, .alpha="Observe")
p2 <- mpse %>% mp_plot_rarecurve(.rare=RareAbundanceRarecurve, .alpha="Observe", .group=time)
p3 <- mpse %>% mp_plot_rarecurve(.rare=RareAbundanceRarecurve, .alpha="Observe", .group=time, plot.group=TRUE)

## End(Not run)
```

---

mp_plot_upset *Plotting the different number of OTU between group via UpSet plot*

Description

Plotting the different number of OTU between group via UpSet plot

Usage

```
mp_plot_upset(.data, .group, .upset = NULL, ...)

## S4 method for signature 'MPSE'
mp_plot_upset(.data, .group, .upset = NULL, ...)

## S4 method for signature 'tbl_mpse'
mp_plot_upset(.data, .group, .upset = NULL, ...)

## S4 method for signature 'grouped_df_mpse'
mp_plot_upset(.data, .group, .upset = NULL, ...)
```

Arguments

.data	MPSE obejct or tbl_mpse object
.group	the column name of group
.upset	the column name of result after run mp_cal_upset
...	additional parameters, see also 'scale_x_upset' of 'ggupset'.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(mouse.time.mpse)
mpse <- mouse.time.mpse %>%
        mp_rrarefy(.abundance=Abundance) %>%
        mp_cal_upset(.abundance=RareAbundance, .group=time)
mpse
p <- mpse %>% mp_plot_upset(.group=time, .upset=ggupsetOftime)
p

## End(Not run)
```

---

mp_plot_venn *Plotting the different number of OTU between groups with Venn Diagram.*

Description

Plotting the different number of OTU between groups with Venn Diagram.

Usage

```
mp_plot_venn(.data, .group, .venn = NULL, ...)

## S4 method for signature 'MPSE'
mp_plot_venn(.data, .group, .venn = NULL, ...)

## S4 method for signature 'tbl_mpse'
mp_plot_venn(.data, .group, .venn = NULL, ...)

## S4 method for signature 'grouped_df_mpse'
mp_plot_venn(.data, .group, .venn = NULL, ...)
```

Arguments

.data	MPSE object or tbl_mpse object
.group	the column names of group to be visualized
.venn	the column names of result after run mp_cal_venn.
...	additional parameters, such as 'size', 'label_size', 'edge_size' etc, see also 'ggVennDiagram'.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(mouse.time.mpse)
mpse <- mouse.time.mpse %>%
        mp_rrarefy() %>%
        mp_cal_venn(.abundance=RareAbundance, .group=time, action="add")
mpse
p <- mpse %>% mp_plot_venn(.group=time, .venn=vennOftime)
p

## End(Not run)
```

---

mp_rrarefy *mp_rrarefy method*

Description

mp_rrarefy method

Usage

```
mp_rrarefy(
  .data,
  .abundance = NULL,
  raresize,
  trimOTU = FALSE,
  trimSample = FALSE,
  seed = 123,
  ...
)

## S4 method for signature 'MPSE'
mp_rrarefy(
  .data,
  .abundance = NULL,
```

```
  raresize,
  trimOTU = FALSE,
  trimSample = FALSE,
  seed = 123,
  ...
)

## S4 method for signature 'tbl_mpse'
mp_rrarefy(
  .data,
  .abundance = NULL,
  raresize,
  trimOTU = FALSE,
  trimSample = FALSE,
  seed = 123,
  ...
)

## S4 method for signature 'grouped_df_mpse'
mp_rrarefy(
  .data,
  .abundance = NULL,
  raresize,
  trimOTU = FALSE,
  trimSample = FALSE,
  seed = 123,
  ...
)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the name of OTU(feature) abundance column, default is Abundance.
raresize	integer Subsample size for rarefying community.
trimOTU	logical Whether to remove the otus that are no longer present in any sample after rarefaction
trimSample	logical whether to remove the samples that do not have enough abundance (raresize), default is FALSE.
seed	a random seed to make the rrarefy reproducible, default is 123.
...	additional parameters, meaningless now.

Value

update object

Author(s)

Shuangbin Xu

See Also

[mp_extract_assays()] and [mp_decostand()]

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %>% mp_rrarefy()
```

`mp_select_as_tip` *select specific taxa level as rownames of MPSE*

Description

select specific taxa level as rownames of MPSE

Usage

```
mp_select_as_tip(x, tip.level = "OTU")

## S4 method for signature 'MPSE'
mp_select_as_tip(x, tip.level = "OTU")

## S4 method for signature 'tbl_mpse'
mp_select_as_tip(x, tip.level = "OTU")

## S4 method for signature 'grouped_df_mpse'
mp_select_as_tip(x, tip.level = "OTU")
```

Arguments

`x`	MPSE object
`tip.level`	the taxonomy level, default is 'OTU'.

Examples

```
## Not run:
data(mouse.time.mpse)
newmpse <- mouse.time.mpse %>%
           mp_select_as_tip(tip.level = Species)
newmpse

## End(Not run)
```

mp_stat_taxa *Count the number and total number taxa for each sample at different taxonomy levels*

Description

Count the number and total number taxa for each sample at different taxonomy levels

Usage

```
mp_stat_taxa(.data, .abundance, action = "add", ...)

## S4 method for signature 'MPSE'
mp_stat_taxa(.data, .abundance, action = "add", ...)

## S4 method for signature 'tbl_mpse'
mp_stat_taxa(.data, .abundance, action = "add", ...)

## S4 method for signature 'grouped_df_mpse'
mp_stat_taxa(.data, .abundance, action = "add", ...)
```

Arguments

.data	MPSE or tbl_mpse object
.abundance	the column name of abundance to be calculated
action	a character "get" returns a table only contained the number and total number for each sample at different taxonomy levels, "only" returns a non-redundant tibble contained a nest column (StatTaxaInfo) and other sample information, "add" returns a update object (.data) contained a nest column (StatTaxaInfo).
...	additional parameter

Value

update object or tbl_df according action argument

Author(s)

Shuangbin Xu

Examples

```
data(mouse.time.mpse)
mouse.time.mpse %>%
 mp_stat_taxa(.abundance=Abundance, action="only")
```

multi_compare *a container for performing two or more sample test.*

Description

a container for performing two or more sample test.

Usage

```
multi_compare(
  fun = wilcox.test,
  data,
  feature,
  factorNames,
  subgroup = NULL,
  ...
)
```

Arguments

fun	character, the method for test, optional ""
data	data.frame, nrow sample * ncol feature+factorNames.
feature	vector, the features wanted to test.
factorNames	character, the name of a factor giving the corresponding groups.
subgroup	vector, the names of groups, default is NULL.
...	additional arguments for fun.

Value

the result of fun, if fun is wilcox.test, it will return the list with class "htest".

Author(s)

Shuangbin Xu

Examples

```
datest <- data.frame(A=rnorm(1:10,mean=5),
                     B=rnorm(2:11, mean=6),
                     group=c(rep("case",5),rep("control",5)))
head(datest)
multi_compare(fun=wilcox.test,data=datest,
              feature=c("A", "B"),factorNames="group")
da2 <- data.frame(A=rnorm(1:15,mean=5),
                  B=rnorm(2:16,mean=6),
                  group=c(rep("case1",5),rep("case2",5),rep("control",5)))
multi_compare(fun=wilcox.test,data=da2,
```

```
feature=c("A", "B"),factorNames="group",
subgroup=c("case1", "case2"))
```

`ordplotClass-class` *ordplotClass class*

Description

ordplotClass class

Slots

`coord` matrix object contained the coordinate for ordination plot.

`xlab` character object contained the text of xlab for ordination plot.

`ylab` character object contained the text of ylab for ordination plot.

`title` character object contained the text of title for ordination plot.

`pcasample-class` *pcasample class*

Description

pcasample class

Slots

`pca` prcomp or pcoa object

`sampleda` associated sample information

`pcoa-class` *pcoa class*

Description

pcoa class

See Also

`pcoa`

prcomp-class *prcomp class*

Description

prcomp class

See Also

prcomp

print *print some objects*

Description

print some objects

Usage

```
## S3 method for class 'MPSE'
print(
  x,
  ...,
  n = NULL,
  width = NULL,
  max_extra_cols = NULL,
  max_footer_lines = NULL
)

## S3 method for class 'tbl_mpse'
print(x, ..., n = NULL, width = NULL, max_extra_cols = NULL)

## S3 method for class 'grouped_df_mpse'
print(x, ..., n = NULL, width = NULL, max_extra_cols = NULL)

## S3 method for class 'rarecurve'
print(x, ..., n = NULL, width = NULL, max_extra_cols = NULL)
```

Arguments

x	Object to format or print.
...	Other arguments passed on to individual methods.
n	Number of rows to show. If ‘NULL‘, the default, will print all rows if less than option ‘tibble.print_max‘. Otherwise, will print ‘tibble.print_min‘ rows.

width	Width of text output to generate. This defaults to 'NULL', which means use 'getOption("tibble.width")' or (if also 'NULL') 'getOption("width")'; the latter displays only the columns that fit on one screen. You can also set 'options(tibble.width = Inf)' to override this default and always print all columns.
max_extra_cols	Number of extra columns to print abbreviated information for, if the width is too small for the entire tibble. If 'NULL', the default, will print information about at most 'tibble.max_extra_cols' extra columns.
max_footer_lines	integer maximum number of lines for the footer.

Value

print information

`read_qza` *read the qza file, output of qiime2.*

Description

the function was designed to read the ouput of qiime2.

Usage

```
read_qza(qzafile, parallel = FALSE)
```

Arguments

qzafile	character, the format of file should be one of 'BIOMV210DirFmt', 'TSVTaxonomyDirectoryFormat', 'NewickDirectoryFormat' and 'DNASequencesDirectoryFormat'.
parallel	logical, whether parsing the taxonomy by multi-parallel, efault is FALSE.

Value

list contained one or multiple object of feature table, taxonomy table, tree and represent sequences.

Examples

```
## Not run:
otuqzafile <- system.file("extdata", "table.qza",
                          package="MicrobiotaProcess")
otuqza <- read_qza(otuqzafile)
str(otuqza)

## End(Not run)
```

reexports *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

dplyr arrange, distinct, filter, group_by, left_join, mutate, pull, rename, select, slice, ungroup

ggplot2 fortify, remove_missing

ggtree td_filter, td_unnest

magrittr %<>%, %>%, extract

SummarizedExperiment colData, colData<-, rowData

tibble as_tibble

tidyr nest, unnest

tidytree as.treedata

scale_fill_diff_cladogram *Create the scale of mp_plot_diff_cladogram.*

Description

Create the scale of mp_plot_diff_cladogram.

Usage

```
scale_fill_diff_cladogram(values, breaks = waiver(), na.value = "grey50", ...)
```

Arguments

values	a set of aesthetic values (different group (default)) to map data values to.
breaks	One of 'NULL' for no breaks, 'waiver()' for the default breaks, A character vector of breaks.
na.value	The aesthetic value to use for missing ('NA') values.
...	see also 'discrete_scale' of 'ggplot2'.

set_diff_boxplot_color

set the color scale of plot generated by mp_plot_diff_boxplot

Description

set the color scale of plot generated by mp_plot_diff_boxplot

Usage

```
set_diff_boxplot_color(.data, values, ...)
```

Arguments

.data	the aplot object generated by mp_plot_diff_boxplot.
values	the color vector, required.
...	additional parameters, see also the 'scale_fill_manual' of 'ggplot2'

set_scale_theme

adjust the color of heatmap of mp_plot_dist

Description

adjust the color of heatmap of mp_plot_dist

Usage

```
set_scale_theme(.data, x, aes_var)
```

Arguments

.data	the plot of heatmap of mp_plot_dist
x	the scale or theme
aes_var	character the variable (column) name of color or size.

show,diffAnalysisClass-method *method extensions to show for diffAnalysisClass or alphasample objects.*

Description

method extensions to show for diffAnalysisClass or alphasample objects.

Usage

```
## S4 method for signature 'diffAnalysisClass'
show(object)

## S4 method for signature 'alphasample'
show(object)

## S4 method for signature 'MPSE'
show(object)
```

Arguments

object	object, diffAnalysisClass or alphasample class

Value

print info

Author(s)

Shuangbin Xu

Examples

```
## Not run:
data(kostic2012crc)
kostic2012crc %<>% as.phyloseq()
head(phyloseq::sample_data(kostic2012crc),3)
kostic2012crc <- phyloseq::rarefy_even_depth(kostic2012crc,rngseed=1024)
table(phyloseq::sample_data(kostic2012crc)$DIAGNOSIS)
set.seed(1024)
diffres <- diff_analysis(kostic2012crc, classgroup="DIAGNOSIS",
                         mlfun="lda", filtermod="fdr",
                         firstcomfun = "kruskal.test",
                         firstalpha=0.05, strictmod=TRUE,
                         secondcomfun = "wilcox.test",
                         subclmin=3, subclwilc=TRUE,
                         secondalpha=0.01, lda=3)
show(diffres)
```

```
## End(Not run)
```

---

split_data *Split Large Vector or DataFrame*

Description

Split large vector or dataframe to list class, which contian subset vectors or dataframe of origin vector or dataframe.

Usage

```
split_data(x, nums, chunks = NULL, random = FALSE)
```

Arguments

x	vector class or data.frame class.
nums	integer.
chunks	integer. use chunks if nums is missing. Note nums and chunks shouldn't concurrently be NULL, default is NULL.
random	bool, whether split randomly, default is FALSE, if you want to split data randomly, you can set TRUE, and if you want the results are reproducible, you should add seed before.

Value

the subset of x, vector or data.frame class.

Author(s)

Shuangbin Xu

Examples

```
data(iris)
irislist <- split_data(iris, 40)
dalist <- c(1:100)
dalist <- split_data(dalist, 30)
```

split_str_to_list *split a dataframe contained one column*

Description

split a dataframe contained one column with a specify field separator character.

Usage

```
split_str_to_list(
  strdataframe,
  prefix = "tax",
  sep = "; ",
  extra = "drop",
  fill = "right",
  ...
)
```

Arguments

strdataframe	dataframe; a dataframe contained one column to split.
prefix	character; the result dataframe columns names prefix, default is "tax".
sep	character; the field separator character, default is "; ".
extra	character; See separate details.
fill	character; See separate details.
...	Additional arguments passed to separate.

Value

data.frame of strdataframe by sep.

Author(s)

Shuangbin Xu

Examples

```
## Not run:
    otudafile <- system.file("extdata", "otu_tax_table.txt",
                          package="MicrobiotaProcess")
    samplefile <- system.file("extdata",
                    "sample_info.txt", package="MicrobiotaProcess")
    otuda <- read.table(otudafile, sep="\t", header=TRUE,
                        row.names=1, check.names=FALSE,
                        skip=1, comment.char="")
    sampleda <- read.table(samplefile,
                sep="\t", header=TRUE, row.names=1)
```

```
    taxdf <- otuda[!sapply(otuda, is.numeric)]
    taxdf <- split_str_to_list(taxdf)
    head(taxdf)

## End(Not run)
```

---

taxonomy *extract the taxonomy annotation in MPSE object*

Description

extract the taxonomy annotation in MPSE object

Usage

```
taxonomy(x, ...)

## S4 method for signature 'MPSE'
taxonomy(x, ...)

## S4 method for signature 'tbl_mpse'
taxonomy(x, ...)

## S4 method for signature 'grouped_df_mpse'
taxonomy(x, ...)

mp_extract_taxonomy(x, ...)

## S4 method for signature 'MPSE'
mp_extract_taxonomy(x, ...)

## S4 method for signature 'tbl_mpse'
mp_extract_taxonomy(x, ...)

## S4 method for signature 'grouped_df_mpse'
mp_extract_taxonomy(x, ...)
```

Arguments

x	MPSE object
...	additional arguments

Value

data.frame contained taxonomy information

data.frame contained taxonomy annotation.

theme_taxbar *theme_taxbar*

Description

theme_taxbar

Usage

```
theme_taxbar(
  axis.text.x = element_text(angle = -45, hjust = 0, size = 8),
  legend.position = "bottom",
  legend.box = "horizontal",
  legend.text = element_text(size = 8),
  legend.title = element_blank(),
  strip.text.x = element_text(size = 12, face = "bold"),
  strip.background = element_rect(colour = "white", fill = "grey"),
  ...
)
```

Arguments

axis.text.x	element_text, x axis tick labels.
legend.position	character, default is "bottom".
legend.box	character, arrangement of legends, default is "horizontal".
legend.text	element_text, legend labels text.
legend.title	element_text, legend title text
strip.text.x	element_text, strip text of x
strip.background	element_rect, the background of x
...	additional parameters

Value

updated ggplot object with new theme

See Also

theme

Examples

```
## Not run:
    library(ggplot2)
    data(test_otu_data)
    test_otu_data %<>% as.phyloseq()
    otubar <- ggbartax(test_otu_data, settheme=FALSE) +
        xlab(NULL) + ylab("relative abundance(%)") +
        theme_taxbar()

## End(Not run)
```

Index