

# Package: MicrobiotaProcess (via r-universe)

July 25, 2024

**Type** Package

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

**Version** 1.17.1

**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, gghalves, 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** 7.3.1

**Repository** <https://bioc.r-universe.dev>

**RemoteUrl** <https://github.com/bioc/MicrobiotaProcess>

**RemoteRef** HEAD

**RemoteSha** 892373ba4c2c1cdbbd23523c1095b8c46c1e0657

## Contents

alphasample-class . . . . .	4
as.MPSE . . . . .	5
as.phyloseq . . . . .	5
as.treedata.taxonomyTable . . . . .	6
build_tree . . . . .	7
convert_to_treedata . . . . .	8
data-hmp_aerobiosis_small . . . . .	9
data-kostic2012crc . . . . .	9
data-test_otu_data . . . . .	9
diffAnalysisClass-class . . . . .	10
diff_analysis . . . . .	10
drop_taxa . . . . .	13
dr_extract . . . . .	14
extract_binary_offspring . . . . .	15
generalizedFC . . . . .	16
get_alphaindex . . . . .	17
get_clust . . . . .	18
get_coord.pcoa . . . . .	20
get_count . . . . .	20
get_dist . . . . .	21
get_mean_median . . . . .	23
get_NRI_NTI . . . . .	24
get_pca . . . . .	25
get_pcoa . . . . .	26
get_pvalue . . . . .	27
get_rarecurve . . . . .	28
get_sampledflist . . . . .	29
get_taxadf . . . . .	30
get_upset . . . . .	31
get_varct.pcoa . . . . .	33
get_vennlist . . . . .	34
ggbartax . . . . .	35
ggbox . . . . .	37
ggclust . . . . .	39
ggdiffbox . . . . .	40
ggdiffclade . . . . .	42
ggdifftaxbar . . . . .	44
ggeffectsize . . . . .	46

ggordpoint . . . . .	48
ggrarecurve . . . . .	50
ImportDada2 . . . . .	52
ImportQiime2 . . . . .	53
mouse.time.mpse . . . . .	54
MPSE . . . . .	54
MPSE-accessors . . . . .	55
MPSE-class . . . . .	58
mp_adonis . . . . .	59
mp_aggregate . . . . .	61
mp_aggregate_clade . . . . .	62
mp_anosim . . . . .	64
mp_balance_clade . . . . .	66
mp_cal_abundance . . . . .	68
mp_cal_alpha . . . . .	71
mp_cal_cca . . . . .	73
mp_cal_clust . . . . .	74
mp_cal_dca . . . . .	76
mp_cal_dist . . . . .	77
mp_cal_divergence . . . . .	80
mp_cal_nmds . . . . .	82
mp_cal_pca . . . . .	84
mp_cal_pcoa . . . . .	86
mp_cal_pd_metric . . . . .	88
mp_cal_rarecurve . . . . .	90
mp_cal_rda . . . . .	92
mp_cal_upset . . . . .	93
mp_cal_venn . . . . .	95
mp_decostand . . . . .	97
mp_diff_analysis . . . . .	98
mp_diff_clade . . . . .	104
mp_dmn . . . . .	109
mp_dmnngroup . . . . .	110
mp_envfit . . . . .	111
mp_extract_abundance . . . . .	113
mp_extract_assays . . . . .	114
mp_extract_dist . . . . .	115
mp_extract_feature . . . . .	116
mp_extract_internal_attr . . . . .	117
mp_extract_rarecurve . . . . .	117
mp_extract_refseq . . . . .	118
mp_extract_sample . . . . .	119
mp_extract_tree . . . . .	119
mp_filter_taxa . . . . .	120
mp_fortify . . . . .	122
mp_import_biom . . . . .	122
mp_import_humann_regroup . . . . .	123
mp_import_metaphlan . . . . .	124

mp_import_qiime . . . . .	125
mp_mantel . . . . .	126
mp_mrpp . . . . .	128
mp_plot_abundance . . . . .	130
mp_plot_alpha . . . . .	134
mp_plot_diff_boxplot . . . . .	136
mp_plot_diff_cladogram . . . . .	138
mp_plot_diff_manhattan . . . . .	140
mp_plot_diff_res . . . . .	142
mp_plot_dist . . . . .	145
mp_plot_ord . . . . .	147
mp_plot_rarecurve . . . . .	150
mp_plot_upset . . . . .	152
mp_plot_venn . . . . .	153
mp_rrarefy . . . . .	154
mp_select_as_tip . . . . .	156
mp_stat_taxa . . . . .	157
multi_compare . . . . .	158
ordplotClass-class . . . . .	159
pcasample-class . . . . .	159
print . . . . .	160
read_qza . . . . .	161
scale_fill_diff_cladogram . . . . .	161
set_diff_boxplot_color . . . . .	162
set_scale_theme . . . . .	162
show,diffAnalysisClass-method . . . . .	163
split_data . . . . .	164
split_str_to_list . . . . .	165
taxonomy . . . . .	166
theme_taxbar . . . . .	167
<b>Index</b>	<b>169</b>

---

alphasample-class	<i>alphasample class</i>
-------------------	--------------------------

---

**Description**

alphasample class

**Slots**

- alpha data.frame contained alpha metrics of samples
- sampleda associated sample information

---

as.MPSE	<i>as.MPSE method</i>
---------	-----------------------

---

**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	<i>convert to phyloseq object.</i>
-------------	------------------------------------

---

**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` object.

`.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	<i>building tree</i>
------------	----------------------

---

## 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 <a href="#">AlignSeqs</a> .

## 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	<i>convert dataframe contained hierarchical relationship or other classes to treedata class</i>
---------------------	---

---

## 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, featuredata, sampledata, taxdata featuredata contained 55 samples (nrow) and 1091 features (ncol) sampledata contained 55 samples from 6 body sites of 10 subjects. taxdata contained 699 taxonomy by 6 rank. This datasets were built from the LEfSe.[http://huttenhower.sph.harvard.edu/webfm\\_send/129](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, published 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, taxa, 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 <a href="#">decostand</a> , 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 subclass 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	<i>generalized fold change</i>
---------------	--------------------------------

---

## 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_alphaindex	<i>alpha index</i>
----------------	--------------------

---

**Description**

calculate the alpha index (Observe,Chao1,Shannon,Simpson) of sample with [diversity](#)

**Usage**

```
get_alphaindex(obj, ...)

## S4 method for signature 'matrix'
get_alphaindex(obj, mindepth, sampled, 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.
sampled	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, sampled = NULL, hclustmethod = "average", ...)
```

## S3 method for class 'data.frame'

```
get_clust(
  obj,
  distmethod = "euclidean",
  taxa_are_rows = FALSE,
  sampled = 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 <a href="#">get_dist</a> .
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 <a href="#">as.phylo</a> .
method	character, the standardization methods for community ecologists, see also <a href="#">decostand</a>

### 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	<i>get ordination coordinates.</i>
----------------	------------------------------------

---

**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	<i>calculate the count or relative abundance of replicate element with a specify column</i>
-----------	---

---

**Description**

Caculate the count or relative abundance of replicate element with a specify 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 numeric dataframe, if featurelist is non't NULL, all columns should be numeric.
featurelist	dataframe; a dataframe contained one character 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 <a href="#">distanceMethodList</a>
taxa_are_rows	logical, default is FALSE.
sampleda	data.frame, nrow sample * ncol factors.
tree	object, the phylo class, see also <a href="#">as.phylo</a> .
method	character, default is hellinger, see also <a href="#">decostand</a>

**Value**

distance class contained 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	<i>get the mean and median of specific feature.</i>
-----------------	---

---

## 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)
featuredata <- merge(featureda, sampleda, by=0)
rownames(featureda) <- as.vector(featureda$Row.names)
featureda$Row.names <- NULL
feameamed <- get_mean_median(datameta=featuredata,
                             feature="p__Actinobacteria",
                             subclass="body_site")
fplot <- ggdiffntaxbar(feameamed, featurename="p__Actinobacteria",
                      classgroup="oxygen_availability", subclass="body_site")

## End(Not run)
```

---

get_NRI_NTI	<i>calculating related phylogenetic alpha metric</i>
-------------	--

---

## Description

calculating related phylogenetic alpha metric

## Usage

```
get_NRI_NTI(obj, ...)

## S4 method for signature 'matrix'
get_NRI_NTI(
  obj,
  mindepth,
  sampled,
  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, sampled, 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.
sampled	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	<i>Performs a principal components analysis</i>
---------	---

---

### Description

Performs a principal components analysis

### Usage

```
get_pca(obj, ...)

## S3 method for class 'data.frame'
get_pca(obj, sampled = 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, see <a href="#">prcomp</a> .
sampled	data.frame, nrow sample * ncol factors.
method	character, the standardization methods for community ecologists. see <a href="#">decostand</a> .

### 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 <a href="#">get_dist</a> .
distmethod	character, the method of distance, see also <a href="#">distance</a>
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 unifracs method, it should be required.
method	character, the standardization method for community ecologists, default is hellinger, if the data has be normlized, it should 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	<i>obtain the result of rare curve</i>
---------------	--

---

### Description

generate the result of rare curve.

### Usage

```
get_rarecurve(obj, ...)
```

```
## S4 method for signature 'data.frame'
get_rarecurve(obj, sampled, 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	<i>Generate random data list from a original data.</i>
------------------	--

---

**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	<i>get the data of specified taxonomy</i>
------------	---

---

**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 <- ggbar(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, sampledata, 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.
sampledata	data.frame, if the obj is data.frame, the sampledata should be provided.
factorNames	character, the column names of factor in sampledata
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	<i>get the contribution of variables</i>
----------------	--

---

## 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	<i>generate a vennlist for VennDiagram</i>
--------------	--

---

## 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 <a href="#">ggplot</a>
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,
  sampled,
  factorNames,
  indexNames,
  geom = "boxplot",
  factorLevels = NULL,
  compare = TRUE,
  testmethod = "wilcox.test",
  signifmap = FALSE,
  p_textsize = 2,
  step_increase = 0.1,
  boxwidth = 0.2,
  facetrnrow = 1,
  controlgroup = NULL,
  comparelist = NULL,
  ...
)

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

**Arguments**

<code>obj</code>	object, <code>alphasample</code> or <code>data.frame</code> (row sample x column features).
<code>factorNames</code>	character, the names of factor contained in <code>sampled</code> .
<code>...</code>	additional arguments, see also <a href="#">stat_signif</a> .
<code>sampled</code>	<code>data.frame</code> , sample information if <code>obj</code> is <code>data.frame</code> , the <code>sampled</code> should be provided.
<code>indexNames</code>	character, the vector character, should be the names of features contained object.
<code>geom</code>	character, "boxplot" or "violin", default is "boxplot".
<code>factorLevels</code>	list, the levels of the factors, default is <code>NULL</code> , 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 <a href="#">stat_signif</a> .
signifmap	logical, whether the pvalue are directly written a annotaion or asterisks are used instead, default is (pvalue) FALSE. see also <a href="#">stat_signif</a> .
p_textsize	numeric, the size of text of pvalue or asterisks, default is 2.
step_increase	numeric, see also <a href="#">stat_signif</a> , default is 0.1.
boxwidth	numeric, the width of boxplot when the geom is 'violin', default is 0.2.
facetrnrow	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 <a href="#">geom_tippoint</a>
layout	character, the layout of tree, see also <a href="#">ggtree</a> .
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 <a href="#">geom_boxplot</a> , 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 visualize 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 originalID 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	<i>plot the clade tree with highlight</i>
-------------	---

---

## 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",
  linewidth = 0.6,
  bg.tree.color = "#bed0d1",
  bg.point.color = "#bed0d1",
  bg.point.stroke = 0.2,
  bg.point.fill = "white",
  skpointsize = 2,
  highlight.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 <a href="#">diff_analysis</a> , 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	logical 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, linewidth=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,
  cololist = NULL,
  ylabel = "relative abundance",
  ...
)

```

### Arguments

obj	object, diffAnalysisClass see also <a href="#">diff_analysis</a> or feMeanMedian class, see also <a href="#">get_mean_median</a> .
...	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.
cololist	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)
ggdiffntaxbar(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

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

### 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,
  sampled = 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 <a href="#">geom_text_repel</a> .
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 <a href="#">geom_text_repel</a> .

**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, sampled, 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

<code>obj</code>	phyloseq, phyloseq class or data.frame shape of data.frame (nrow sample * ncol feature ( + factor)).
<code>...</code>	additional parameters, see also <a href="#">ggplot2</a> {ggplot}.
<code>chunks</code>	integer, the number of subsample in a sample, default is 400.
<code>factorLevels</code>	list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.
<code>sampleda</code>	data.frame, (nrow sample * ncol factor)
<code>indexNames</code>	character, default is "Observe", only for "Observe", "Chao1", "ACE".
<code>linesize</code>	integer, default is 0.5.
<code>facetnrow</code>	integer, the nrow of facet, default is 1.
<code>shadow</code>	logical, whether merge samples with group (factorNames) and display the ribbon of group, default is TRUE.
<code>factorNames</code>	character, default is missing.
<code>se</code>	logical, default is FALSE.
<code>method</code>	character, default is lm.
<code>formula</code>	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 output of `dada2`, and generated the `phyloseq` object 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**

<code>seqtab</code>	matrix, feature table, the output of <a href="#">removeBimeraDenovo</a> .
<code>taxatab</code>	matrix, a taxonomic table, the output of <a href="#">assignTaxonomy</a> , or the output of <a href="#">addSpecies</a> .
<code>reftree</code>	phylo, treedata or character, the treedata or phylo class of tree, or the tree file.
<code>sampleda</code>	data.frame or character, the data.frame of sample information, or the file of sample information, nrow samples X ncol factors.
<code>...</code>	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,
  taxaqa = NULL,
  mapfilename = NULL,
  refseqqza = NULL,
  treeqza = NULL,
  parallel = FALSE,
  ...
)

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

## Arguments

otuqza	character, the file contained otu table, the output of qiime2.
taxaqa	character, the file contained taxonomy, the output 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, taxaqa=taxaqzafile,
                       mapfilename=mapfile)
mpse

```

---

mouse.time.mipse	<i>(Data) An example data</i>
------------------	-------------------------------

---

**Description**

This is a MPSE object example data.

---

MPSE	<i>Construct a MPSE object</i>
------	--------------------------------

---

**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. <code>list(Abundance=xx1, RareAbundance=xx2)</code> .
colData	An optional <code>DataFrame</code> describing the samples.
otutree	A <code>treedata</code> object of <code>tidytree</code> package, the result parsed by the functions of <code>treeio</code> .
taxatree	A <code>treedata</code> object of <code>tidytree</code> package, the result parsed by the functions of <code>treeio</code> .
refseq	A <code>XStingSet</code> object of <code>Biostrings</code> package, the result parsed by the <code>readDNASTringSet</code> or <code>readAAStringSet</code> of <code>Biostrings</code> .
...	additional parameters, see also the usage of <a href="#">SummarizedExperiment</a> .

**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	<i>MPSE class</i>
------------	-------------------

---

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	<i>Permutational Multivariate Analysis of Variance Using Distance Matrices for MPSE or tbl_mpse object</i>
-----------	--

---

**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	<i>aggregate the assays with the specific group of sample and fun.</i>
--------------	--

---

## 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 <a href="#">aggregate</a> .

## 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	<i>calculate the mean/median (relative) abundance of internal nodes according to their children tips.</i>
--------------------	---

---

## 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

<code>.data</code>	MPSE object which must contain otutree slot, required
<code>.abundance</code>	the column names of abundance.
<code>force</code>	logical whether calculate the (relative) abundance forcibly when the abundance is not be rarefied, default is FALSE.
<code>relative</code>	logical whether calculate the relative abundance.
<code>aggregate_fun</code>	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'.
<code>action</code>	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.
<code>...</code>	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 function.
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	<i>Calculating the balance score of internal nodes (clade) according to the geometric.mean/mean/median abundance of their binary children tips.</i>
------------------	---

---

## 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	<i>Calculate the (relative) abundance of each taxonomy class for each sample or group.</i>
------------------	--

---

## 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 visualize 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, and '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)
library(gghalves)
p <- ggplot(data=tbl, aes(x=time, y=alpha, fill=time)) +
  geom_half_violin(color=NA, side="l", trim=FALSE) +
  geom_boxplot(aes(color=time), fill=NA, position=position_nudge(x=.22), width=0.2) +
  geom_half_point(side="r", shape=21) +
  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	<i>[Partial] [Constrained] Correspondence Analysis with MPSE or tbl_mpse object</i>
------------	---

---

## 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	<i>Hierarchical cluster analysis for the samples with MPSE or tbl_mpse object</i>
--------------	---

---

**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	<i>Calculate the distances between the samples or features with specified abundance.</i>
-------------	--

---

### 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

<code>.data</code>	MPSE or <code>tbl_mpse</code> object
<code>.abundance</code>	the name of otu abundance to be calculated
<code>.env</code>	the column names of continuous environment factors, default is <code>NULL</code> .
<code>distmethod</code>	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 <code>vegdist</code> of <code>vegan</code> ), and "w", "-1", "c", "wb", "r", "l", "e", "t", "me", "j", "sor", "m", "-2", "co", "cc", "g", "-3", "l", "l9", "hk", "rlb", "sim", "gl", "z" (implemented in <code>betadiver</code> of <code>vegan</code> ), "maximum", "binary", "minkowski" (implemented in <code>dist</code> of <code>stats</code> ), "unifrac", "weighted unifrac" (implemented in <code>phyloseq</code> ), "cor", "abscor", "cosangle", "abscosangle" (implemented in <code>hopach</code> ), or other customized distance function.
<code>action</code>	character, "add" joins the distance data to the object, "only" return a non-redundant tibble with the distance information. "get" return 'dist' object.
<code>scale</code>	logical whether scale the metric of environment ( <code>.env</code> is provided) before the distance was calculated, default is <code>FALSE</code> . The environment matrix can be processed when it was joined to the MPSE or <code>tbl_mpse</code> object.
<code>cal.feature.dist</code>	logical whether to calculate the distance between the features. default is <code>FALSE</code> , meaning calculate the distance between the samples.
<code>...</code>	additional parameters. some dot arguments if <code>distmethod</code> is <code>unifrac</code> or <code>weighted unifrac</code> : <ul style="list-style-type: none"> <li>• <code>weighted</code> logical, whether to use <code>weighted-UniFrac</code> calculation, which considers the relative abundance of taxa, default is <code>FALSE</code>, meaning <code>unweightrd-UniFrac</code>, which only considers presence/absence of taxa.</li> <li>• <code>normalized</code> logical, whether normaized the branch length of tree to the range between 0 and 1 when the <code>weighted=TRUE</code>.</li> <li>• <code>parallel</code> logical, whether to execute the calculation in parallel, default is <code>FALSE</code>.</li> </ul>

## 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	<i>calculate the divergence with MPSE or tbl_mpse</i>
-------------------	---

---

## 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**

<code>.data</code>	MPSE or <code>tbl_mpse</code> object
<code>.abundance</code>	the name of abundance to be calculated.
<code>.dim</code>	integer The number of dimensions to be returned, default is 3.
<code>action</code>	character "add" joins the pca result to the object, "only" return a non-redundant tibble with the pca result. "get" return 'prcomp' object.
<code>...</code>	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	<i>Calculating related phylogenetic alpha metric with MPSE or tbl_mpse object</i>
------------------	---

---

## 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

<code>.data</code>	object, MPSE or <code>tbl_mpse</code> object
<code>.abundance</code>	The column name of OTU abundance column to be calculate.
<code>action</code>	character it has three options, "add" joins the new information to the input <code>tbl</code> (default), "only" return a non-redundant tibble with the just new information, and 'get' return a 'alphasample' object.
<code>metric</code>	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').
<code>abundance.weighted</code>	logical, whether calculate mean nearest taxon distances for each species weighted by species abundance, default is TRUE.
<code>force</code>	logical whether calculate the alpha index even the '.abundance' is not rarefied, default is FALSE.
<code>seed</code>	integer a random seed to make the result reproducible, default is 123.
<code>...</code>	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

<code>.data</code>	MPSE or <code>tbl_mpse</code> object
<code>.abundance</code>	the name of otu abundance to be calculated.
<code>action</code>	character it has three options, "add" joins the new information to the input <code>tbl</code> (default), "only" return a non-redundant tibble with the just new information, and 'get' return a 'rarecurve' object.
<code>chunks</code>	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.
<code>seed</code>	a random seed to make the result reproducible, default is 123.
<code>force</code>	logical whether calculate rarecurve forcibly when the ' <code>.abundance</code> ' is not be rarefied, default is FALSE
<code>...</code>	additional parameters.

### Value

update rarecurve 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	<i>[Partial] [Constrained] Redundancy Analysis with MPSE or tbl_mpse object</i>
------------	---

---

**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 function.

... 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	<i>Calculating the samples or groups for each OTU, the result can be visualized by 'ggupset'</i>
--------------	--

---

### 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

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

<code>.data</code>	MPSE or <code>tbl_mpse</code> object
<code>.group</code>	the name of group to be calculated. if it is no provided, the sample will be used.
<code>.abundance</code>	the name of otu abundance to be calculated. if it is null, the rarefied abundance will be used.
<code>action</code>	character, "add" joins the new information to the tibble of <code>tbl_mpse</code> or <code>rowData</code> of MPSE. "only" and "get" return a non-redundant tibble with the just new information.
<code>force</code>	logical whether calculate the relative abundance forcibly when the abundance is not be rarefied, default is FALSE.
<code>...</code>	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 Provides Several Standardization Methods for Community Data*


---

**Description**

This Function Provides 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**

<code>.data</code>	MPSE or <code>tbl_mpse</code> object
<code>.abundance</code>	the names of otu abundance to be applied standardization.
<code>method</code>	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 <a href="#">decostand</a>
<code>logbase</code>	numeric The logarithm base used in 'method=log', default is 2.
<code>...</code>	additional parameters, see also <a href="#">decostand</a>

**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 $fdr \leq .first.test.alpha$ will be kept, if it is set to pvalue, meanings the features that $pvalue \leq .first.test.alpha$ 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,
    highlight.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	<i>Differential internal and tip nodes (clades) analysis for MPSE or tbl_mpse object</i>
---------------	--

---

## 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 $fdr \leq .first.test.alpha$ will be kept, if it is set to pvalue, meanings the features that $pvalue \leq .first.test.alpha$ 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 than 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 <a href="#">mclapply</a> and <a href="#">dmn</a> .

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	<i>Dirichlet-Multinomial generative classifiers to MPSE or tbl_mpse</i>
-------------	---

---

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 <a href="#">mclapply</a> and <a href="#">dmngroup</a> .

**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	<i>Fits an Environmental Vector or Factor onto an Ordination With MPSE or tbl_mpse Object</i>
-----------	---

---

**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=A1, .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=A1, 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)

```

**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	<i>extract the abundance matrix from MPSE object or tbl_mpse object</i>
-------------------	---

---

**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	<i>extract the dist object from MPSE or tbl_mpse object</i>
-----------------	---

---

### 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'.

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

### Value

dist object or `tbl_df` object when `.group` is provided.

---

<code>mp_extract_feature</code>	<i>extract the feature (OTU) information in MPSE object</i>
---------------------------------	---

---

### 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

<code>x</code>	MPSE object
<code>addtaxa</code>	logical whether adding the taxonomy information default is <code>FALSE</code> .
<code>...</code>	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 ggrrarecurve

---

mp_extract_refseq	<i>Extract the representative sequences from MPSE object</i>
-------------------	--

---

**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	<i>extract the sample information in MPSE object</i>
-------------------	--

---

**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	<i>extract the taxonomy tree in MPSE object</i>
-----------------	---

---

**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	<i>Filter OTU (Features) By Abundance Level</i>
----------------	---

---

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)
```

---

<code>mp_fortify</code>	<i>mp_fortify</i>
-------------------------	-------------------

---

**Description**

Fortify a model with data in MicrobiotaProcess

**Usage**

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

**Arguments**

<code>model</code>	object
<code>...</code>	additional parameters

**Value**

data frame or tbl\_df object

---

<code>mp_import_biom</code>	<i>building MPSE object from biom-format file.</i>
-----------------------------	--

---

**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

---

<code>mp_import_metaphlan</code>	<i>Import function to load the output of MetaPhlAn.</i>
----------------------------------	---

---

**Description**

Import function to load the output of MetaPhlAn.

**Usage**

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

**Arguments**

<code>profile</code>	the output file (text format) of MetaPhlAn.
<code>mapfilename</code>	the sample information file or data.frame, default is NULL.
<code>treefile</code>	the path of MetaPhlAn tree file ( mpa_v30_CHOCOPhlAn_201901_species_tree.nwk), default is NULL.
<code>linenum</code>	<p>a integer, sometimes the output file of MetaPhlAn ( &lt; 3) contained the sample information in the first several lines. The linenum should be required. for example:</p> <pre>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 ...</pre> <p>the linenum should be set to 3.</p> <pre>sampleid A1 A2 A3 A4 A5 Bacteria 99 99 99 99 99 ...</pre> <p>The linenum should be set to 1.</p>
<code>...</code>	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	<i>Import function to load the output of qiime.</i>
-----------------	---

---

**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 output 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 <a href="#">mantel</a> .

**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	<i>plotting the abundance of taxa via specified taxonomy class</i>
-------------------	--

---

**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

<code>.data</code>	MPSE object or <code>tbl_mpse</code> object
<code>.abundance</code>	the column name of abundance to be plotted.
<code>.group</code>	the column name of group to be calculated and plotted, default is <code>NULL</code> .
<code>taxa.class</code>	name of taxonomy class, default is <code>NULL</code> , meaning the Phylum class will be plotted.
<code>topn</code>	integer the number of the top most abundant, default is 10.
<code>relative</code>	logical whether calculate the relative abundance and plotted.
<code>force</code>	logical whether calculate the relative abundance forcibly when the abundance is not be rarefied, default is <code>FALSE</code> .
<code>plot.group</code>	logical whether plotting the abundance of specified <code>taxa.class</code> taxonomy with group not sample level, default is <code>FALSE</code> .
<code>geom</code>	character which type plot, options is 'flowbar' 'bar' and 'heatmap', default is 'flowbar'.
<code>feature.dist</code>	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 [ <code>mp_cal_dist()</code> ] (except <code>unifrac</code> related).
<code>feature.hclust</code>	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

<code>.data</code>	MPSE or <code>tbl_mpse</code> object
<code>.group</code>	the column name of sample group information
<code>.alpha</code>	the column name of alpha index after run <code>mp_cal_alpha</code> or <code>mp_cal_pd_metric</code> .
<code>test</code>	the name of the statistical test, default is 'wilcox.test'
<code>comparisons</code>	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 <code>NULL</code> , meaning it will be calculated automatically with the names in the <code>.group</code> .
<code>step_increase</code>	numeric vector with the increase in fraction of total height for every additional comparison to minimize overlap, default is 0.05.
<code>...</code>	additional parameters, see also <a href="#">geom_signif</a>

### 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",
  highlight.alpha = 0.3,
  highlight.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

<code>.data</code>	MPSE object or treedata which was from the taxatree slot after running the 'mp_diff_analysis'.
<code>.group</code>	the column name for mapping the different color.
<code>.size</code>	the column name for mapping the size of points, default is 'pvalue'.
<code>taxa.class</code>	the taxonomy class name will be replaced shorthand, default is the one level above 'OTU'.
<code>removeUnknown</code>	logical, whether mask the unknown taxonomy information but differential species, default is FALSE.
<code>layout</code>	character, the layout of tree, default is 'radial', see also the 'layout' of 'ggtree'.
<code>hilight.alpha</code>	numeric, the transparency of high light clade, default is 0.3.
<code>hilight.size</code>	numeric, the margin thickness of high light clade, default is 0.2.
<code>bg.tree.size</code>	numeric, the line size (width) of tree, default is 0.15.
<code>bg.tree.color</code>	character, the line color of tree, default is '#bed0d1'.
<code>bg.point.color</code>	character, the color of margin of background node points of tree, default is '#bed0d1'.
<code>bg.point.fill</code>	character, the point fill (since point shape is 21) of background nodes of tree, default is 'white'.
<code>bg.point.stroke</code>	numeric, the margin thickness of point of background nodes of tree, default is 0.2.
<code>bg.point.size</code>	numeric, the point size of background nodes of tree, default is 2.
<code>label.size</code>	numeric, the label size of differential taxa, default is 2.6.
<code>tip.annot</code>	logical whether to replace the differential tip labels with shorthand, default is TRUE.
<code>as.tiplab</code>	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'.
<code>...</code>	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,
    highlight.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	<i>Plotting the distance between the samples with heatmap or boxplot.</i>
--------------	---

---

## 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

<code>.data</code>	the MPSE or <code>tbl_mpse</code> object after <code>[mp_cal_dist()]</code> is performed with <code>action="add"</code>
<code>.distmethod</code>	the column names of distance of samples, it will generate after <code>[mp_cal_dist()]</code> is performed.
<code>.group</code>	the column names of group, default is <code>NULL</code> , when it is not provided the heatmap of distance between samples will be returned. If it is provided and <code>group.test</code> is <code>TURE</code> , the comparisons boxplot of distance between the group will be returned, but when <code>group.test</code> is <code>FALSE</code> , the heatmap of distance between samples with group information will be returned.
<code>group.test</code>	logical default is <code>FALSE</code> , see the <code>.group</code> argument.
<code>hclustmethod</code>	character the method of <a href="#">hclust</a> , default is 'average' (= UPGMA).
<code>test</code>	the name of the statistical test, default is 'wilcox.test'
<code>comparisons</code>	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 <code>NULL</code> , meaning it will be calculated automatically with the names in the <code>.group</code> .
<code>step_increase</code>	numeric vector with the increase in fraction of total height for every additional comparison to minimize overlap, default is 0.1.
<code>...</code>	additional parameters, see also <a href="#">geom_signif</a>

### 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  $p \leq 0.05$  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=A1, .size=Mn)
p1
p2 <- mpse %>% mp_plot_ord(.ord=CCA, .group=A1, .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=A1, .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 <a href="#">facet_wrap</a> .
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 <a href="#">geom_smooth</a> .

**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 object 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	<i>Plotting the different number of OTU between groups with Venn Diagram.</i>
--------------	---

---

**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)
```

---

<i>mp_rrarefy</i>	<i>mp_rrarefy method</i>
-------------------	--------------------------

---

**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 (rare-size), 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	<i>select specific taxa level as rownames of MPSE</i>
------------------	---

---

**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	<i>Count the number and total number taxa for each sample at different taxonomy levels</i>
--------------	--

---

## Description

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

## Usage

```
mp_stat_taxa(.data, .abundance, action = "add", ...)\n\n## S4 method for signature 'MPSE'\nmp_stat_taxa(.data, .abundance, action = "add", ...)\n\n## S4 method for signature 'tbl_mpse'\nmp_stat_taxa(.data, .abundance, action = "add", ...)\n\n## S4 method for signature 'grouped_df_mpse'\nmp_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)\nmouse.time.mpse %>%\n  mp_stat_taxa(.abundance=Abundance, action="only")
```

---

multi_compare	<i>a container for performing two or more sample test.</i>
---------------	--

---

## 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	<i>ordplotClass class</i>
--------------------	---------------------------

---

**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	<i>pcasample class</i>
-----------------	------------------------

---

**Description**

pcasample class

**Slots**

- pca precomp or pcoa object
- sampleda associated sample information

---

<code>print</code>	<i>print some objects</i>
--------------------	---------------------------

---

**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**

<code>x</code>	Object to format or print.
<code>...</code>	Other arguments passed on to individual methods.
<code>n</code>	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.
<code>width</code>	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.
<code>max_extra_cols</code>	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.
<code>max_footer_lines</code>	integer maximum number of lines for the footer.

**Value**

print information



---

read_qza	<i>read the qza file, output of qiime2.</i>
----------	---

---

**Description**

the function was designed to read the output of qiime2.

**Usage**

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

**Arguments**

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

**Value**

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

**Examples**

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

## End(Not run)
```

---

scale_fill_diff_cladogram	<i>Create the scale of mp_plot_diff_cladogram.</i>
---------------------------	--

---

**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 contain 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	<i>split a dataframe contained one column</i>
-------------------	---

---

## 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 <a href="#">separate</a> details.
fill	character; See <a href="#">separate</a> details.
...	Additional arguments passed to <a href="#">separate</a> .

## 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 <- ggbarntax(test_otu_data, settheme=FALSE) +
  xlab(NULL) + ylab("relative abundance(%)") +
  theme_taxbar()

## End(Not run)
```



# Index

## \* data

- data-hmp\_aerobiosis\_small, [9](#)
- data-kostic2012cnc, [9](#)
- data-test\_otu\_data, [9](#)
- mouse.time.mpse, [54](#)
- [,MPSE,ANY,ANY,ANY-method  
(MPSE-accessors), [55](#)
- addSpecies, [52](#)
- aggregate, [61](#)
- AlignSeqs, [7](#)
- alphasample-class, [4](#)
- as.MPSE, [5](#)
- as.mpse (as.MPSE), [5](#)
- as.phylo, [19](#), [22](#)
- as.phyloseq, [5](#)
- as.treedata.taxonomyTable, [6](#)
- as\_phyloseq (as.phyloseq), [5](#)
- assignTaxonomy, [52](#)
- build\_tree, [7](#)
- build\_tree,character (build\_tree), [7](#)
- build\_tree,character-method  
(build\_tree), [7](#)
- build\_tree,DNABin (build\_tree), [7](#)
- build\_tree,DNABin-method (build\_tree), [7](#)
- build\_tree,DNAStringSet (build\_tree), [7](#)
- build\_tree,DNAStringSet-method  
(build\_tree), [7](#)
- colData<-,MPSE,DataFrame-method  
(MPSE-accessors), [55](#)
- colData<-,MPSE,NULL-method  
(MPSE-accessors), [55](#)
- convert\_to\_treedata, [8](#)
- data-hmp\_aerobiosis\_small, [9](#)
- data-kostic2012cnc, [9](#)
- data-test\_otu\_data, [9](#)
- decostand, [11](#), [19](#), [22](#), [25](#), [98](#)

- diff\_analysis, [10](#), [10](#), [42](#), [45](#), [47](#)
- diffAnalysisClass-class, [10](#)
- distance, [22](#), [26](#)
- distanceMethodList, [22](#)
- diversity, [17](#)
- dmn, [109](#)
- dmngroup, [110](#)
- dr\_extract, [14](#)
- drop\_taxa, [13](#)
- drop\_taxa,data.frame (drop\_taxa), [13](#)
- drop\_taxa,data.frame-method  
(drop\_taxa), [13](#)
- drop\_taxa,phyloseq (drop\_taxa), [13](#)
- drop\_taxa,phyloseq-method (drop\_taxa),  
[13](#)
- extract\_binary\_offspring, [15](#)
- facet\_wrap, [152](#)
- generalizedFC, [16](#)
- geom\_boxplot, [41](#)
- geom\_signif, [135](#), [146](#)
- geom\_smooth, [152](#)
- geom\_text\_repel, [49](#)
- geom\_tippoint, [39](#)
- get\_alphaindex, [17](#)
- get\_alphaindex,data.frame  
(get\_alphaindex), [17](#)
- get\_alphaindex,data.frame-method  
(get\_alphaindex), [17](#)
- get\_alphaindex,integer  
(get\_alphaindex), [17](#)
- get\_alphaindex,integer-method  
(get\_alphaindex), [17](#)
- get\_alphaindex,matrix (get\_alphaindex),  
[17](#)
- get\_alphaindex,matrix-method  
(get\_alphaindex), [17](#)

get\_alphaindex, numeric  
     (get\_alphaindex), 17  
 get\_alphaindex, numeric-method  
     (get\_alphaindex), 17  
 get\_alphaindex, phyloseq  
     (get\_alphaindex), 17  
 get\_alphaindex, phyloseq-method  
     (get\_alphaindex), 17  
 get\_clust, 18  
 get\_coord (get\_coord.pcoa), 20  
 get\_coord.pcoa, 20  
 get\_count, 20  
 get\_dist, 19, 21, 26  
 get\_mean\_median, 23, 45  
 get\_NRI\_NTI, 24  
 get\_NRI\_NTI, data.frame (get\_NRI\_NTI), 24  
 get\_NRI\_NTI, data.frame-method  
     (get\_NRI\_NTI), 24  
 get\_NRI\_NTI, matrix (get\_NRI\_NTI), 24  
 get\_NRI\_NTI, matrix-method  
     (get\_NRI\_NTI), 24  
 get\_NRI\_NTI, phyloseq (get\_NRI\_NTI), 24  
 get\_NRI\_NTI, phyloseq-method  
     (get\_NRI\_NTI), 24  
 get\_pca, 25  
 get\_pcoa, 26  
 get\_pvalue, 27  
 get\_rarecurve, 28  
 get\_rarecurve, data.frame  
     (get\_rarecurve), 28  
 get\_rarecurve, data.frame-method  
     (get\_rarecurve), 28  
 get\_rarecurve, phyloseq (get\_rarecurve), 28  
 get\_rarecurve, phyloseq-method  
     (get\_rarecurve), 28  
 get\_ratio (get\_count), 20  
 get\_sampledflist, 29  
 get\_taxadf, 30  
 get\_taxadf, data.frame (get\_taxadf), 30  
 get\_taxadf, data.frame-method  
     (get\_taxadf), 30  
 get\_taxadf, phyloseq (get\_taxadf), 30  
 get\_taxadf, phyloseq-method  
     (get\_taxadf), 30  
 get\_upset, 31  
 get\_upset, data.frame (get\_upset), 31  
 get\_upset, data.frame-method  
     (get\_upset), 31  
 get\_upset, phyloseq (get\_upset), 31  
 get\_upset, phyloseq-method (get\_upset), 31  
 get\_varct (get\_varct.pcoa), 33  
 get\_varct.pcoa, 33  
 get\_vennlist, 34  
 get\_vennlist, data.frame-method  
     (get\_vennlist), 34  
 get\_vennlist, data.frame  
     (get\_vennlist), 34  
 get\_vennlist, phyloseq (get\_vennlist), 34  
 get\_vennlist, phyloseq-method  
     (get\_vennlist), 34  
 ggbartax, 35  
 ggbartaxa (ggbartax), 35  
 ggbox, 37  
 ggbox, alphasample (ggbox), 37  
 ggbox, alphasample-method (ggbox), 37  
 ggbox, data.frame (ggbox), 37  
 ggbox, data.frame-method (ggbox), 37  
 ggclust, 39  
 ggdiffbartaxa (ggdiffntaxbar), 44  
 ggdiffbox, 40  
 ggdiffbox, diffAnalysisClass  
     (ggdiffbox), 40  
 ggdiffbox, diffAnalysisClass-method  
     (ggdiffbox), 40  
 ggdiffclade, 42  
 ggdiffntaxbar, 44  
 ggdiffntaxbar, diffAnalysisClass  
     (ggdiffntaxbar), 44  
 ggdiffntaxbar, diffAnalysisClass-method  
     (ggdiffntaxbar), 44  
 ggdiffntaxbar.featureMeanMedian  
     (ggdiffntaxbar), 44  
 ggeffectsize, 46  
 ggordpoint, 48  
 ggplot, 36  
 ggplot2, 51  
 ggrarecurve, 50  
 ggtree, 39  
  
 hclust, 146  
 hmp\_aerobiosis\_small  
     (data-hmp\_aerobiosis\_small), 9  
  
 import\_dada2 (ImportDada2), 52  
 import\_qiime2 (ImportQiime2), 53

- ImportDada2, [52](#)
- ImportQiime2, [53](#)
- kostic2012crc (data-kostic2012crc), [9](#)
- mantel, [128](#)
- mclapply, [109](#), [110](#)
- mouse.time.mpse, [54](#)
- mp\_adonis, [59](#)
- mp\_adonis,grouped\_df\_mpse (mp\_adonis), [59](#)
- mp\_adonis,grouped\_df\_mpse-method (mp\_adonis), [59](#)
- mp\_adonis,MPSE (mp\_adonis), [59](#)
- mp\_adonis,MPSE-method (mp\_adonis), [59](#)
- mp\_adonis,tbl\_mpse (mp\_adonis), [59](#)
- mp\_adonis,tbl\_mpse-method (mp\_adonis), [59](#)
- mp\_aggregate, [61](#)
- mp\_aggregate,MPSE (mp\_aggregate), [61](#)
- mp\_aggregate,MPSE-method (mp\_aggregate), [61](#)
- mp\_aggregate\_clade, [62](#)
- mp\_aggregate\_clade,grouped\_df\_mpse (mp\_aggregate\_clade), [62](#)
- mp\_aggregate\_clade,grouped\_df\_mpse-method (mp\_aggregate\_clade), [62](#)
- mp\_aggregate\_clade,MPSE (mp\_aggregate\_clade), [62](#)
- mp\_aggregate\_clade,MPSE-method (mp\_aggregate\_clade), [62](#)
- mp\_aggregate\_clade,tbl\_mpse (mp\_aggregate\_clade), [62](#)
- mp\_aggregate\_clade,tbl\_mpse-method (mp\_aggregate\_clade), [62](#)
- mp\_anosim, [64](#)
- mp\_anosim,grouped\_df\_mpse (mp\_anosim), [64](#)
- mp\_anosim,grouped\_df\_mpse-method (mp\_anosim), [64](#)
- mp\_anosim,MPSE (mp\_anosim), [64](#)
- mp\_anosim,MPSE-method (mp\_anosim), [64](#)
- mp\_anosim,tbl\_mpse (mp\_anosim), [64](#)
- mp\_anosim,tbl\_mpse-method (mp\_anosim), [64](#)
- mp\_balance\_clade, [66](#)
- mp\_balance\_clade,grouped\_df\_mpse (mp\_balance\_clade), [66](#)
- mp\_balance\_clade,MPSE (mp\_balance\_clade), [66](#)
- mp\_balance\_clade,MPSE-method (mp\_balance\_clade), [66](#)
- mp\_balance\_clade,tbl\_mpse (mp\_balance\_clade), [66](#)
- mp\_balance\_clade,tbl\_mpse-method (mp\_balance\_clade), [66](#)
- mp\_cal\_abundance, [68](#)
- mp\_cal\_abundance,grouped\_df\_mpse (mp\_cal\_abundance), [68](#)
- mp\_cal\_abundance,grouped\_df\_mpse-method (mp\_cal\_abundance), [68](#)
- mp\_cal\_abundance,MPSE (mp\_cal\_abundance), [68](#)
- mp\_cal\_abundance,MPSE-method (mp\_cal\_abundance), [68](#)
- mp\_cal\_abundance,tbl\_mpse (mp\_cal\_abundance), [68](#)
- mp\_cal\_abundance,tbl\_mpse-method (mp\_cal\_abundance), [68](#)
- mp\_cal\_alpha, [71](#)
- mp\_cal\_alpha,grouped\_df\_mpse (mp\_cal\_alpha), [71](#)
- mp\_cal\_alpha,grouped\_df\_mpse-method (mp\_cal\_alpha), [71](#)
- mp\_cal\_alpha,MPSE (mp\_cal\_alpha), [71](#)
- mp\_cal\_alpha,MPSE-method (mp\_cal\_alpha), [71](#)
- mp\_cal\_alpha,tbl\_mpse (mp\_cal\_alpha), [71](#)
- mp\_cal\_alpha,tbl\_mpse-method (mp\_cal\_alpha), [71](#)
- mp\_cal\_cca, [73](#)
- mp\_cal\_cca,grouped\_df\_mpse (mp\_cal\_cca), [73](#)
- mp\_cal\_cca,grouped\_df\_mpse-method (mp\_cal\_cca), [73](#)
- mp\_cal\_cca,MPSE (mp\_cal\_cca), [73](#)
- mp\_cal\_cca,MPSE-method (mp\_cal\_cca), [73](#)
- mp\_cal\_cca,tbl\_mpse (mp\_cal\_cca), [73](#)
- mp\_cal\_cca,tbl\_mpse-method (mp\_cal\_cca), [73](#)
- mp\_cal\_clust, [74](#)
- mp\_cal\_clust,grouped\_df\_mpse (mp\_cal\_clust), [74](#)
- mp\_cal\_clust,grouped\_df\_mpse-method (mp\_cal\_clust), [74](#)

- (mp\_cal\_clust), 74
- mp\_cal\_clust, MPSE (mp\_cal\_clust), 74
- mp\_cal\_clust, MPSE-method (mp\_cal\_clust), 74
- mp\_cal\_clust, tbl\_mpse (mp\_cal\_clust), 74
- mp\_cal\_clust, tbl\_mpse-method (mp\_cal\_clust), 74
- mp\_cal\_dca, 76
- mp\_cal\_dca, grouped\_df\_mpse (mp\_cal\_dca), 76
- mp\_cal\_dca, grouped\_df\_mpse-method (mp\_cal\_dca), 76
- mp\_cal\_dca, MPSE (mp\_cal\_dca), 76
- mp\_cal\_dca, MPSE-method (mp\_cal\_dca), 76
- mp\_cal\_dca, tbl\_mpse (mp\_cal\_dca), 76
- mp\_cal\_dca, tbl\_mpse-method (mp\_cal\_dca), 76
- mp\_cal\_dist, 77
- mp\_cal\_dist, grouped\_df\_mpse (mp\_cal\_dist), 77
- mp\_cal\_dist, grouped\_df\_mpse-method (mp\_cal\_dist), 77
- mp\_cal\_dist, MPSE (mp\_cal\_dist), 77
- mp\_cal\_dist, MPSE-method (mp\_cal\_dist), 77
- mp\_cal\_dist, tbl\_mpse (mp\_cal\_dist), 77
- mp\_cal\_dist, tbl\_mpse-method (mp\_cal\_dist), 77
- mp\_cal\_divergence, 80
- mp\_cal\_divergence, grouped\_df\_mpse (mp\_cal\_divergence), 80
- mp\_cal\_divergence, grouped\_df\_mpse-method (mp\_cal\_divergence), 80
- mp\_cal\_divergence, MPSE (mp\_cal\_divergence), 80
- mp\_cal\_divergence, MPSE-method (mp\_cal\_divergence), 80
- mp\_cal\_divergence, tbl\_mpse (mp\_cal\_divergence), 80
- mp\_cal\_divergence, tbl\_mpse-method (mp\_cal\_divergence), 80
- mp\_cal\_nmds, 82
- mp\_cal\_nmds, grouped\_df\_mpse (mp\_cal\_nmds), 82
- mp\_cal\_nmds, grouped\_df\_mpse-method (mp\_cal\_nmds), 82
- mp\_cal\_nmds, MPSE (mp\_cal\_nmds), 82
- mp\_cal\_nmds, MPSE-method (mp\_cal\_nmds), 82
- mp\_cal\_pca, 84
- mp\_cal\_pca, grouped\_df\_mpse (mp\_cal\_pca), 84
- mp\_cal\_pca, grouped\_df\_mpse-method (mp\_cal\_pca), 84
- mp\_cal\_pca, MPSE (mp\_cal\_pca), 84
- mp\_cal\_pca, MPSE-method (mp\_cal\_pca), 84
- mp\_cal\_pca, tbl\_mpse (mp\_cal\_pca), 84
- mp\_cal\_pca, tbl\_mpse-method (mp\_cal\_pca), 84
- mp\_cal\_pcoa, 86
- mp\_cal\_pcoa, grouped\_df\_mpse (mp\_cal\_pcoa), 86
- mp\_cal\_pcoa, grouped\_df\_mpse-method (mp\_cal\_pcoa), 86
- mp\_cal\_pcoa, MPSE (mp\_cal\_pcoa), 86
- mp\_cal\_pcoa, MPSE-method (mp\_cal\_pcoa), 86
- mp\_cal\_pcoa, tbl\_mpse (mp\_cal\_pcoa), 86
- mp\_cal\_pcoa, tbl\_mpse-method (mp\_cal\_pcoa), 86
- mp\_cal\_pd\_metric, 88
- mp\_cal\_pd\_metric, grouped\_df\_mpse (mp\_cal\_pd\_metric), 88
- mp\_cal\_pd\_metric, grouped\_df\_mpse-method (mp\_cal\_pd\_metric), 88
- mp\_cal\_pd\_metric, MPSE (mp\_cal\_pd\_metric), 88
- mp\_cal\_pd\_metric, MPSE-method (mp\_cal\_pd\_metric), 88
- mp\_cal\_pd\_metric, tbl\_mpse (mp\_cal\_pd\_metric), 88
- mp\_cal\_pd\_metric, tbl\_mpse-method (mp\_cal\_pd\_metric), 88
- mp\_cal\_rarecurve, 90
- mp\_cal\_rarecurve, grouped\_df\_mpse (mp\_cal\_rarecurve), 90
- mp\_cal\_rarecurve, grouped\_df\_mpse-method (mp\_cal\_rarecurve), 90
- mp\_cal\_rarecurve, MPSE (mp\_cal\_rarecurve), 90
- mp\_cal\_rarecurve, MPSE-method (mp\_cal\_rarecurve), 90
- mp\_cal\_rarecurve, tbl\_mpse

- (mp\_cal\_rarecurve), 90
- mp\_cal\_rarecurve, tbl\_mpse-method  
(mp\_cal\_rarecurve), 90
- mp\_cal\_rda, 92
- mp\_cal\_rda, grouped\_df\_mpse  
(mp\_cal\_rda), 92
- mp\_cal\_rda, grouped\_df\_mpse-method  
(mp\_cal\_rda), 92
- mp\_cal\_rda, MPSE (mp\_cal\_rda), 92
- mp\_cal\_rda, MPSE-method (mp\_cal\_rda), 92
- mp\_cal\_rda, tbl\_mpse (mp\_cal\_rda), 92
- mp\_cal\_rda, tbl\_mpse-method  
(mp\_cal\_rda), 92
- mp\_cal\_upset, 93
- mp\_cal\_upset, grouped\_df\_mpse  
(mp\_cal\_upset), 93
- mp\_cal\_upset, grouped\_df\_mpse-method  
(mp\_cal\_upset), 93
- mp\_cal\_upset, MPSE (mp\_cal\_upset), 93
- mp\_cal\_upset, MPSE-method  
(mp\_cal\_upset), 93
- mp\_cal\_upset, tbl\_mpse (mp\_cal\_upset), 93
- mp\_cal\_upset, tbl\_mpse-method  
(mp\_cal\_upset), 93
- mp\_cal\_venn, 95
- mp\_cal\_venn, grouped\_df\_mpse  
(mp\_cal\_venn), 95
- mp\_cal\_venn, grouped\_df\_mpse-method  
(mp\_cal\_venn), 95
- mp\_cal\_venn, MPSE (mp\_cal\_venn), 95
- mp\_cal\_venn, MPSE-method (mp\_cal\_venn),  
95
- mp\_cal\_venn, tbl\_mpse (mp\_cal\_venn), 95
- mp\_cal\_venn, tbl\_mpse-method  
(mp\_cal\_venn), 95
- mp\_decostand, 97
- mp\_decostand, data.frame (mp\_decostand),  
97
- mp\_decostand, data.frame-method  
(mp\_decostand), 97
- mp\_decostand, grouped\_df\_mpse  
(mp\_decostand), 97
- mp\_decostand, grouped\_df\_mpse-method  
(mp\_decostand), 97
- mp\_decostand, MPSE (mp\_decostand), 97
- mp\_decostand, MPSE-method  
(mp\_decostand), 97
- mp\_decostand, tbl\_mpse (mp\_decostand), 97
- mp\_decostand, tbl\_mpse-method  
(mp\_decostand), 97
- mp\_diff\_analysis, 98
- mp\_diff\_analysis, grouped\_df\_mpse  
(mp\_diff\_analysis), 98
- mp\_diff\_analysis, grouped\_df\_mpse-method  
(mp\_diff\_analysis), 98
- mp\_diff\_analysis, MPSE  
(mp\_diff\_analysis), 98
- mp\_diff\_analysis, MPSE-method  
(mp\_diff\_analysis), 98
- mp\_diff\_analysis, tbl\_mpse  
(mp\_diff\_analysis), 98
- mp\_diff\_analysis, tbl\_mpse-method  
(mp\_diff\_analysis), 98
- mp\_diff\_clade, 104
- mp\_diff\_clade, grouped\_df\_mpse  
(mp\_diff\_clade), 104
- mp\_diff\_clade, grouped\_df\_mpse-method  
(mp\_diff\_clade), 104
- mp\_diff\_clade, MPSE (mp\_diff\_clade), 104
- mp\_diff\_clade, MPSE-method  
(mp\_diff\_clade), 104
- mp\_diff\_clade, tbl\_mpse (mp\_diff\_clade),  
104
- mp\_diff\_clade, tbl\_mpse-method  
(mp\_diff\_clade), 104
- mp\_dmn, 109
- mp\_dmn, grouped\_df\_mpse (mp\_dmn), 109
- mp\_dmn, grouped\_df\_mpse-method (mp\_dmn),  
109
- mp\_dmn, MPSE (mp\_dmn), 109
- mp\_dmn, MPSE-method (mp\_dmn), 109
- mp\_dmn, tbl\_mpse (mp\_dmn), 109
- mp\_dmn, tbl\_mpse-method (mp\_dmn), 109
- mp\_dmngroup, 110
- mp\_dmngroup, grouped\_df\_mpse  
(mp\_dmngroup), 110
- mp\_dmngroup, grouped\_df\_mpse-method  
(mp\_dmngroup), 110
- mp\_dmngroup, MPSE (mp\_dmngroup), 110
- mp\_dmngroup, MPSE-method (mp\_dmngroup),  
110
- mp\_dmngroup, tbl\_mpse (mp\_dmngroup), 110
- mp\_dmngroup, tbl\_mpse-method  
(mp\_dmngroup), 110
- mp\_envfit, 111
- mp\_envfit, grouped\_df\_mpse (mp\_envfit),

- 111
- mp\_envfit, grouped\_df\_mpse-method  
(mp\_envfit), 111
- mp\_envfit, MPSE (mp\_envfit), 111
- mp\_envfit, MPSE-method (mp\_envfit), 111
- mp\_envfit, tbl\_mpse (mp\_envfit), 111
- mp\_envfit, tbl\_mpse-method (mp\_envfit),  
111
- mp\_extract\_abundance, 113
- mp\_extract\_abundance, grouped\_df\_mpse  
(mp\_extract\_abundance), 113
- mp\_extract\_abundance, grouped\_df\_mpse-method  
(mp\_extract\_abundance), 113
- mp\_extract\_abundance, MPSE  
(mp\_extract\_abundance), 113
- mp\_extract\_abundance, MPSE-method  
(mp\_extract\_abundance), 113
- mp\_extract\_abundance, tbl\_mpse  
(mp\_extract\_abundance), 113
- mp\_extract\_abundance, tbl\_mpse-method  
(mp\_extract\_abundance), 113
- mp\_extract\_assays, 114
- mp\_extract\_assays, grouped\_df\_mpse  
(mp\_extract\_assays), 114
- mp\_extract\_assays, grouped\_df\_mpse-method  
(mp\_extract\_assays), 114
- mp\_extract\_assays, MPSE  
(mp\_extract\_assays), 114
- mp\_extract\_assays, MPSE-method  
(mp\_extract\_assays), 114
- mp\_extract\_assays, tbl\_mpse  
(mp\_extract\_assays), 114
- mp\_extract\_assays, tbl\_mpse-method  
(mp\_extract\_assays), 114
- mp\_extract\_dist, 115
- mp\_extract\_dist, grouped\_df\_mpse  
(mp\_extract\_dist), 115
- mp\_extract\_dist, grouped\_df\_mpse-method  
(mp\_extract\_dist), 115
- mp\_extract\_dist, MPSE (mp\_extract\_dist),  
115
- mp\_extract\_dist, MPSE-method  
(mp\_extract\_dist), 115
- mp\_extract\_dist, tbl\_mpse  
(mp\_extract\_dist), 115
- mp\_extract\_dist, tbl\_mpse-method  
(mp\_extract\_dist), 115
- mp\_extract\_feature, 116
- mp\_extract\_feature, grouped\_df\_mpse  
(mp\_extract\_feature), 116
- mp\_extract\_feature, grouped\_df\_mpse-method  
(mp\_extract\_feature), 116
- mp\_extract\_feature, MPSE  
(mp\_extract\_feature), 116
- mp\_extract\_feature, MPSE-method  
(mp\_extract\_feature), 116
- mp\_extract\_feature, tbl\_mpse  
(mp\_extract\_feature), 116
- mp\_extract\_feature, tbl\_mpse-method  
(mp\_extract\_feature), 116
- mp\_extract\_internal\_attr, 117
- mp\_extract\_internal\_attr, grouped\_df\_mpse  
(mp\_extract\_internal\_attr), 117
- mp\_extract\_internal\_attr, grouped\_df\_mpse-method  
(mp\_extract\_internal\_attr), 117
- mp\_extract\_internal\_attr, MPSE  
(mp\_extract\_internal\_attr), 117
- mp\_extract\_internal\_attr, MPSE-method  
(mp\_extract\_internal\_attr), 117
- mp\_extract\_internal\_attr, tbl\_mpse  
(mp\_extract\_internal\_attr), 117
- mp\_extract\_internal\_attr, tbl\_mpse-method  
(mp\_extract\_internal\_attr), 117
- mp\_extract\_otutree (mp\_extract\_tree),  
119
- mp\_extract\_rarecurve, 117
- mp\_extract\_rarecurve, grouped\_df\_mpse  
(mp\_extract\_rarecurve), 117
- mp\_extract\_rarecurve, grouped\_df\_mpse-method  
(mp\_extract\_rarecurve), 117
- mp\_extract\_rarecurve, MPSE  
(mp\_extract\_rarecurve), 117
- mp\_extract\_rarecurve, MPSE-method  
(mp\_extract\_rarecurve), 117
- mp\_extract\_rarecurve, tbl\_mpse  
(mp\_extract\_rarecurve), 117
- mp\_extract\_rarecurve, tbl\_mpse-method  
(mp\_extract\_rarecurve), 117
- mp\_extract\_refseq, 118
- mp\_extract\_refseq, grouped\_df\_mpse  
(mp\_extract\_refseq), 118
- mp\_extract\_refseq, grouped\_df\_mpse-method  
(mp\_extract\_refseq), 118
- mp\_extract\_refseq, MPSE  
(mp\_extract\_refseq), 118
- mp\_extract\_refseq, MPSE-method

- (mp\_extract\_refseq), 118
- mp\_extract\_refseq, tbl\_mpse
  - (mp\_extract\_refseq), 118
- mp\_extract\_refseq, tbl\_mpse-method
  - (mp\_extract\_refseq), 118
- mp\_extract\_sample, 119
- mp\_extract\_sample, grouped\_df\_mpse
  - (mp\_extract\_sample), 119
- mp\_extract\_sample, grouped\_df\_mpse-method
  - (mp\_extract\_sample), 119
- mp\_extract\_sample, MPSE
  - (mp\_extract\_sample), 119
- mp\_extract\_sample, MPSE-method
  - (mp\_extract\_sample), 119
- mp\_extract\_sample, tbl\_mpse
  - (mp\_extract\_sample), 119
- mp\_extract\_sample, tbl\_mpse-method
  - (mp\_extract\_sample), 119
- mp\_extract\_taxatree (mp\_extract\_tree), 119
- mp\_extract\_taxonomy (taxonomy), 166
- mp\_extract\_taxonomy, grouped\_df\_mpse
  - (taxonomy), 166
- mp\_extract\_taxonomy, grouped\_df\_mpse-method
  - (taxonomy), 166
- mp\_extract\_taxonomy, MPSE (taxonomy), 166
- mp\_extract\_taxonomy, MPSE-method
  - (taxonomy), 166
- mp\_extract\_taxonomy, tbl\_mpse
  - (taxonomy), 166
- mp\_extract\_taxonomy, tbl\_mpse-method
  - (taxonomy), 166
- mp\_extract\_tree, 119
- mp\_extract\_tree, grouped\_df\_mpse
  - (mp\_extract\_tree), 119
- mp\_extract\_tree, grouped\_df\_mpse-method
  - (mp\_extract\_tree), 119
- mp\_extract\_tree, MPSE (mp\_extract\_tree), 119
- mp\_extract\_tree, MPSE-method
  - (mp\_extract\_tree), 119
- mp\_extract\_tree, tbl\_mpse
  - (mp\_extract\_tree), 119
- mp\_extract\_tree, tbl\_mpse-method
  - (mp\_extract\_tree), 119
- mp\_filter\_taxa, 120
- mp\_filter\_taxa, grouped\_df\_mpse
  - (mp\_filter\_taxa), 120
- mp\_filter\_taxa, MPSE (mp\_filter\_taxa), 120
- mp\_filter\_taxa, MPSE-method
  - (mp\_filter\_taxa), 120
- mp\_filter\_taxa, tbl\_mpse
  - (mp\_filter\_taxa), 120
- mp\_filter\_taxa, tbl\_mpse-method
  - (mp\_filter\_taxa), 120
- mp\_fortify, 122
- mp\_import\_biom, 122
- mp\_import\_dada2 (ImportDada2), 52
- mp\_import\_humann\_regroup, 123
- mp\_import\_metaphlan, 124
- mp\_import\_qiime, 125
- mp\_import\_qiime2 (ImportQiime2), 53
- mp\_mantel, 126
- mp\_mantel, grouped\_df\_mpse (mp\_mantel), 126
- mp\_mantel, grouped\_df\_mpse-method
  - (mp\_mantel), 126
- mp\_mantel, MPSE (mp\_mantel), 126
- mp\_mantel, MPSE-method (mp\_mantel), 126
- mp\_mantel, tbl\_mpse (mp\_mantel), 126
- mp\_mantel, tbl\_mpse-method (mp\_mantel), 126
- mp\_mrpp, 128
- mp\_mrpp, grouped\_df\_mpse (mp\_mrpp), 128
- mp\_mrpp, grouped\_df\_mpse-method
  - (mp\_mrpp), 128
- mp\_mrpp, MPSE (mp\_mrpp), 128
- mp\_mrpp, MPSE-method (mp\_mrpp), 128
- mp\_mrpp, tbl\_mpse (mp\_mrpp), 128
- mp\_mrpp, tbl\_mpse-method (mp\_mrpp), 128
- mp\_plot\_abundance, 130
- mp\_plot\_abundance, grouped\_df\_mpse
  - (mp\_plot\_abundance), 130
- mp\_plot\_abundance, grouped\_df\_mpse-method
  - (mp\_plot\_abundance), 130
- mp\_plot\_abundance, MPSE
  - (mp\_plot\_abundance), 130
- mp\_plot\_abundance, MPSE-method
  - (mp\_plot\_abundance), 130
- mp\_plot\_abundance, tbl\_mpse
  - (mp\_plot\_abundance), 130
- mp\_plot\_abundance, tbl\_mpse-method
  - (mp\_plot\_abundance), 130







- mp\_plot\_upset, tbl\_mpse (mp\_plot\_upset), 152
- mp\_plot\_upset, tbl\_mpse-method (mp\_plot\_upset), 152
- mp\_plot\_venn, 153
- mp\_plot\_venn, grouped\_df\_mpse (mp\_plot\_venn), 153
- mp\_plot\_venn, grouped\_df\_mpse-method (mp\_plot\_venn), 153
- mp\_plot\_venn, MPSE (mp\_plot\_venn), 153
- mp\_plot\_venn, MPSE-method (mp\_plot\_venn), 153
- mp\_plot\_venn, tbl\_mpse (mp\_plot\_venn), 153
- mp\_plot\_venn, tbl\_mpse-method (mp\_plot\_venn), 153
- mp\_rrarefy, 154
- mp\_rrarefy, grouped\_df\_mpse (mp\_rrarefy), 154
- mp\_rrarefy, grouped\_df\_mpse-method (mp\_rrarefy), 154
- mp\_rrarefy, MPSE (mp\_rrarefy), 154
- mp\_rrarefy, MPSE-method (mp\_rrarefy), 154
- mp\_rrarefy, tbl\_mpse (mp\_rrarefy), 154
- mp\_rrarefy, tbl\_mpse-method (mp\_rrarefy), 154
- mp\_select\_as\_tip, 156
- mp\_select\_as\_tip, grouped\_df\_mpse (mp\_select\_as\_tip), 156
- mp\_select\_as\_tip, grouped\_df\_mpse-method (mp\_select\_as\_tip), 156
- mp\_select\_as\_tip, MPSE (mp\_select\_as\_tip), 156
- mp\_select\_as\_tip, MPSE-method (mp\_select\_as\_tip), 156
- mp\_select\_as\_tip, tbl\_mpse (mp\_select\_as\_tip), 156
- mp\_select\_as\_tip, tbl\_mpse-method (mp\_select\_as\_tip), 156
- mp\_stat\_taxa, 157
- mp\_stat\_taxa, grouped\_df\_mpse (mp\_stat\_taxa), 157
- mp\_stat\_taxa, grouped\_df\_mpse-method (mp\_stat\_taxa), 157
- mp\_stat\_taxa, MPSE (mp\_stat\_taxa), 157
- mp\_stat\_taxa, MPSE-method (mp\_stat\_taxa), 157
- mp\_stat\_taxa, tbl\_mpse (mp\_stat\_taxa), 157
- mp\_stat\_taxa, tbl\_mpse-method (mp\_stat\_taxa), 157
- MPSE, 54
- MPSE-accessors, 55
- MPSE-class, 58
- multi\_compare, 158
- ordplotClass-class, 159
- otutree (MPSE-accessors), 55
- otutree, group\_df\_mpse (MPSE-accessors), 55
- otutree, MPSE (MPSE-accessors), 55
- otutree, MPSE-method (MPSE-accessors), 55
- otutree, tbl\_mpse (MPSE-accessors), 55
- otutree, tbl\_mpse-method (MPSE-accessors), 55
- otutree<- (MPSE-accessors), 55
- otutree<-, grouped\_df\_mpse (MPSE-accessors), 55
- otutree<-, grouped\_df\_mpse, NULL-method (MPSE-accessors), 55
- otutree<-, grouped\_df\_mpse, treedata-method (MPSE-accessors), 55
- otutree<-, MPSE (MPSE-accessors), 55
- otutree<-, MPSE, NULL-method (MPSE-accessors), 55
- otutree<-, MPSE, phylo-method (MPSE-accessors), 55
- otutree<-, MPSE, treedata-method (MPSE-accessors), 55
- otutree<-, tbl\_mpse (MPSE-accessors), 55
- otutree<-, tbl\_mpse, NULL-method (MPSE-accessors), 55
- otutree<-, tbl\_mpse, treedata-method (MPSE-accessors), 55
- p.adjust, 150
- pcasample-class, 159
- prcomp, 25
- print, 160
- read\_qza, 161
- refsequence (MPSE-accessors), 55
- refsequence, MPSE (MPSE-accessors), 55
- refsequence, MPSE-method (MPSE-accessors), 55
- refsequence<- (MPSE-accessors), 55
- refsequence<-, MPSE (MPSE-accessors), 55

refsequence<- ,MPSE, NULL-method  
     (MPSE-accessors), 55  
 refsequence<- ,MPSE, XStringSet-method  
     (MPSE-accessors), 55  
 removeBimeraDenovo, 52  
 rownames<- ,MPSE (MPSE-accessors), 55  
 rownames<- ,MPSE-method  
     (MPSE-accessors), 55  
  
 scale\_fill\_diff\_cladogram, 161  
 separate, 165  
 set\_diff\_boxplot\_color, 162  
 set\_scale\_theme, 162  
 show, alphasample-method  
     (show, diffAnalysisClass-method),  
     163  
 show, diffAnalysisClass-method, 163  
 show, MPSE-method  
     (show, diffAnalysisClass-method),  
     163  
 split\_data, 164  
 split\_str\_to\_list, 165  
 stat\_ellipse, 150  
 stat\_signif, 37, 38  
 SummarizedExperiment, 55, 58  
  
 tax\_table (MPSE-accessors), 55  
 tax\_table, grouped\_df\_mpse  
     (MPSE-accessors), 55  
 tax\_table, grouped\_df\_mpse-method  
     (MPSE-accessors), 55  
 tax\_table, MPSE (MPSE-accessors), 55  
 tax\_table, MPSE-method (MPSE-accessors),  
     55  
 tax\_table, tbl\_mpse (MPSE-accessors), 55  
 tax\_table, tbl\_mpse-method  
     (MPSE-accessors), 55  
 taxatree (MPSE-accessors), 55  
 taxatree, grouped\_df\_mpse  
     (MPSE-accessors), 55  
 taxatree, grouped\_df\_mpse-method  
     (MPSE-accessors), 55  
 taxatree, MPSE (MPSE-accessors), 55  
 taxatree, MPSE-method (MPSE-accessors),  
     55  
 taxatree, tbl\_mpse (MPSE-accessors), 55  
 taxatree, tbl\_mpse-method  
     (MPSE-accessors), 55  
 taxatree<- (MPSE-accessors), 55  
  
 taxatree<- ,grouped\_df\_mpse  
     (MPSE-accessors), 55  
 taxatree<- ,grouped\_df\_mpse, NULL-method  
     (MPSE-accessors), 55  
 taxatree<- ,grouped\_df\_mpse, treedata-method  
     (MPSE-accessors), 55  
 taxatree<- ,MPSE (MPSE-accessors), 55  
 taxatree<- ,MPSE, NULL-method  
     (MPSE-accessors), 55  
 taxatree<- ,MPSE, treedata-method  
     (MPSE-accessors), 55  
 taxatree<- ,tbl\_mpse (MPSE-accessors), 55  
 taxatree<- ,tbl\_mpse, NULL-method  
     (MPSE-accessors), 55  
 taxatree<- ,tbl\_mpse, treedata-method  
     (MPSE-accessors), 55  
 taxonomy, 166  
 taxonomy, grouped\_df\_mpse (taxonomy), 166  
 taxonomy, grouped\_df\_mpse-method  
     (taxonomy), 166  
 taxonomy, MPSE (taxonomy), 166  
 taxonomy, MPSE-method (taxonomy), 166  
 taxonomy, tbl\_mpse (taxonomy), 166  
 taxonomy, tbl\_mpse-method (taxonomy), 166  
 taxonomy<- (MPSE-accessors), 55  
 taxonomy<- ,MPSE (MPSE-accessors), 55  
 taxonomy<- ,MPSE, data.frame-method  
     (MPSE-accessors), 55  
 taxonomy<- ,MPSE, matrix-method  
     (MPSE-accessors), 55  
 taxonomy<- ,MPSE, NULL-method  
     (MPSE-accessors), 55  
 taxonomy<- ,MPSE, taxonomyTable-method  
     (MPSE-accessors), 55  
 test\_otu\_data (data-test\_otu\_data), 9  
 theme, 167  
 theme\_taxbar, 167