Package ‘MicrobiotaProcess’

March 30, 2021

Type Package

Title an R package for analysis, visualization and biomarker discovery of microbiome

Version 1.2.1

Description MicrobiotaProcess is an R package for analysis, visualization and biomarker discovery of microbial datasets. It supports calculating alpha index and provides functions to visualize rarefaction curves. Moreover, it also supports visualizing the abundance of taxonomy of samples. And It also provides functions to perform the PCA, PCoA and hierarchical cluster analysis. In addition, MicrobiotaProcess also provides a method for the biomarker discovery of metagenome or other datasets.

Depends R (>= 4.0.0)

Imports ape, plyr, tidyr, ggplot2, phyloseq, magrittr, dplyr, Biostrings, ggrepel, vegan, reshape, zoo, ggtree, tidytree, gtools, MASS, methods, rlang, tibble, grDevices, stats, utils, coin, ggsignif, Rmisc, patchwork, ggstar

Suggests DT, prettydoc, treeio, tidyverse, testthat, knitr, nlme, phangorn, DECIPHER, randomForest, biomformat, scales, yaml

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

LazyData false

biocViews Visualization, Microbiome, Software, MultipleComparison, FeatureExtraction

RoxygenNote 7.1.1

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

git_branch RELEASE_3_12

git_last_commit 9140254

git_last_commit_date 2021-03-23

Date/Publication 2021-03-29

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

Maintainer Shuangbin Xu <xshuangbin@163.com>

R topics documented:

`alphasample-class` *alphasample class*

Description

alphasample class

Slots

`alpha` data.frame contained alpha metrics of samples

`sampleda` associated sample information

`as.data.frame.diffAnalysisClass`
get the table of diffAnalysisClass

Description

get the table of diffAnalysisClass

Usage

```
## S3 method for class 'diffAnalysisClass'
as.data.frame(x, ...)

## S3 method for class 'alphasample'
as.data.frame(x, ...)

## S3 method for class 'diffAnalysisClass'
as.data.frame(x, ...)

## S3 method for class 'alphasample'
as.data.frame(x, ...)
```

Arguments

`x`	object, diffAnalysisClass
`...,`	additional parameters

Value

a data.frame contained results of diff_analysis

Examples

```
data(kostic2012crc)
kostic2012crc
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)
restab <- as.data.frame(diffres)
head(restab)
```

---

as.treedata.taxonomyTable *as.treedata*

Description

convert taxonomyTable to treedata

Usage

```
## S3 method for class 'taxonomyTable'
as.treedata(tree, ...)
```

Arguments

tree	object, This is for taxonomyTable class, so it should be a taxonomyTable.
...	additional parameters.

Examples

```
data(test_otu_data)
tree <- as.treedata(phyloseq::tax_table(test_otu_data))
```

---

build_tree *building tree*

Description

The function can be used to building tree.

Usage

```
build_tree(seqs, ...)

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

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

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

Arguments

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

Value

the phylo class of tree.

Author(s)

Shuangbin Xu

Examples

```
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)
```

`clustplotClass-class` *clustplotClass class*

Description

clustplotClass class

Slots

`hclustphylo` phylo object (convert hclust to phylo).

`sampleda` assocaited sample information.

`distmethod` character the method of dist.

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

Description

convert dataframe contained hierarchical relationship or other classes to treedata class

Usage

```
convert_to_treedata(data, type = "species", ...)
```

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".
...,	additional parameters.

Value

treedata class.

Author(s)

Shuangbin Xu

Examples

```
data(hmp_aerobiosis_small)
head(taxda)
treedat <- convert_to_treedata(taxda)
```

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

Description

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

Examples

```
data(hmp_aerobiosis_small)
```

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

Description

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

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

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

Arguments

`obj`	object,a phyloseq class contained otu_table, sample_data, taxda, or data.frame, nrow sample * ncol features.
`...,`	additional parameters.

sampleda — data.frame, nrow sample * ncol factor, the sample names of sampleda and data should be the same.

classgroup — character, the factor name in sampleda.

subclass — character, the factor name in sampleda, default is NULL, meaning no subclass compare.

taxda — data.frame, the classification of the feature in data. default is NULL.

alltax — logical, whether to set all classification as features if taxda is not NULL, default is TRUE.

standard_method — character, the method of standardization, see also decostand, default is NULL, it represents that the relative abundance of taxonomy will be used. If count was set, it represents the count reads of taxonomy will be used.

mlfun — character, the method for calculating the effect size of features, choose "lda" or "rf", default is "lda".

ratio — numeric, range from 0 to 1, the proportion of samples for calculating the effect size of features, default is 0.7.

firstcomfun — character, the method for first test, "oneway.test" for normal distributions, suggested choosing "kruskal.test" for uneven distributions, default is "kruskal.test", or you can use lm, glm, or glm.nb (for negative binomial distribution), or ‘kruskal_test‘, ‘oneway_test‘ of ‘coin‘.

padjust — character, the correction method, default is "fdr".

filtermod — character, the method to filter, default is "pvalue".

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

strictmod — logical, whether to performed in one-against-one, default is TRUE (strict).

fcfun — character, default is "generalizedFC", it can’t be set another at the present time.

secondcomfun — character, the method for one-against-one, default is "wilcox.test" for uneven distributions, or ‘wilcox_test‘ of ‘coin‘, or you can also use ‘lm‘, ‘glm‘, ‘glm.nb‘(for negative binomial distribution in ‘MASS‘).

clmin — integer, the minimum number of samples per classgroup for performing test, default is 5.

clwilc — logical, whether to perform test of per classgroup, default is TRUE.

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

subclmin — integer, the minimum number of samples per suclass for performing test, default is 3.

subclwilc — logical, whether to perform test of per subclass, default is TRUE, meaning more strict.

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

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

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

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

type — character, the type of datasets, default is "species", if the dataset is not about species, such as dataset of kegg function, you should set it to "others".

Value

diff_analysis class.

Author(s)

Shuangbin Xu

Examples

```
data(kostic2012crc)
kostic2012crc
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)
```

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

```
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)
keepps <- drop_taxa(test_otu_data, minocc=0.1, minabu=0)
```

generalizedFC *generalized fold change*

Description

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

Usage

```
generalizedFC(x, ...)

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

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

Arguments

`x` numeric vector, numeric vector of data values or formula, example 'Ozone ~ Month', Ozone is a numeric variable giving the data values 'Month' a factor giving the corresponding groups.

`...` additional arguments.

`y` numeric vector, numeric vector of data values

`base` a positive or complex number, the base with respect to which logarithms are computed, default is 10.

`steps` positive numeric, increment of the sequence, default is 0.05.

`pseudo` positive numeric, avoid the zero for logarithmic, default is 0.00001.

`data` data.frame, an optional matrix or data frame,containing the variables in the formula.

`subset` (similar: see 'wilcox.test')an optional vector specifying a subset of observations to be used.

`na.action` a function which indicates what should happen when the data, contain 'NA's. Defaults to 'getOption("na.action")'.

Value

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

Author(s)

Shuangbin Xu

Examples

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

get_alphaindex *alpha index*

Description

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

Usage

```
get_alphaindex(obj, ...)

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

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

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

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

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

Arguments

`obj`	object, data.frame of (nrow sample * ncol taxonomy(feature)) or phyloseq.
`...`	additional arguments.
`mindepth`	numeric, Subsample size for rarefying community.
`sampleda`	data.frame,sample information, row sample * column factors.

Value

data.frame contained alpha Index.

Author(s)

Shuangbin Xu

Examples

```
library(tidyverse)
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)
set.seed(1024)
alphatab2 <- get_alphaindex(test_otu_data)
head(as.data.frame(alphatab2))
```

`get_clust` *Hierarchical cluster analysis for the samples*

Description

Hierarchical cluster analysis for the samples

Usage

```
get_clust(obj, ...)

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

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

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

Arguments

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

Value

clustplotClass object.

Author(s)

Shuangbin Xu

Examples

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

get_coord.pcoa *get ordination coordinates.*

Description

get ordination coordinates.

Usage

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

get_coord(obj, pc)

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

Arguments

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

Value

ordplotClass object.

Examples

```
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))
```

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

Description

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

Usage

```
get_count(data, featurelist)

get_ratio(data, featurelist)
```

Arguments

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

`featurelist` dataframe; a dataframe contained one chatacter column, default is NULL.

Value

mean of data.frame by featurelist

Author(s)

Shuangbin Xu

Examples

```
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])
```

`get_dist` *calculate distance*

Description

calculate distance

Usage

```
get_dist(obj, ...)

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

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

Arguments

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

Value

distance class contianed distmethod and originalD attr

See Also

`distance`

Examples

```
data(test_otu_data)
distclass <- get_dist(test_otu_data)
hcsample <- get_clust(distclass)
```

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

Description

get the mean and median of specific feature.

Usage

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

Arguments

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

Value

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

Author(s)

Shuangbin Xu

Examples

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

---

get_pca *Performs a principal components analysis*

Description

Performs a principal components analysis

Usage

```
get_pca(obj, ...)

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

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

Arguments

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

Value

pcasample class, contained prcomp class and sample information.

Examples

```
# don't run in examples
#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)
```

---

get_pcoa *performs principal coordinate analysis (PCoA)*

Description

performs principal coordinate analysis (PCoA)

Usage

```
get_pcoa(obj, ...)

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

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

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

Arguments

`obj`	phyloseq, the phyloseq class or dist class.
`...,`	additional parameter, see also `get_dist`.
`distmethod`	character, the method of distance, see also `distance`

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

Value

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

Author(s)

Shuangbin Xu

Examples

```
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)
```

get_pvalue *Methods for computation of the p-value*

Description

Methods for computation of the p-value

Usage

```
get_pvalue(obj)

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

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

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

## S3 method for class 'ScalarIndependenceTest'
```

```
get_pvalue(obj)

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

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

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

Arguments

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

Value

pvalue.

Author(s)

Shuangbin Xu

Examples

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

get_rarecurve *obtain the result of rare curve*

Description

generate the result of rare curve.

Usage

```
get_rarecurve(obj, ...)

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

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

Arguments

`obj` phyloseq class or data.frame shape of data.frame (nrow sample * ncol feature)

`...,` additional parameters.

`sampleda` data.frame, (nrow sample * ncol factor)

`factorLevels` list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.

`chunks` integer, the number of subsample in a sample, default is 400.

Details

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

Value

rarecurve class, which can be visualized by ggrarecurve

Author(s)

Shuangbin Xu

Examples

```
data(test_otu_data)
set.seed(1024)
res <- get_rarecurve(test_otu_data, chunks=200)
p <- ggrarecurve(obj=res,
                 indexNames=c("Observe","Chao1","ACE"),
                 shadow=FALSE,
                 factorNames="Group")
```

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

Description

Generate random data list from a original data.

Usage

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

Arguments

`dalist` list, a list contained multi data.frame.

`bootnums` integer, the number of bootstrap iteration, default is 30.

`ratio` numeric, the ratios of each data.frame to keep.

`makerownames` logical, whether build row.names,default is FALSE.

Value

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

Author(s)

Shuangbin Xu

Examples

```
data(iris)
irislist <- split(iris, iris$Species)
set.seed(1024)
irislist <- get_sampledflist(irislist)
```

`get_taxadf` *get the data of specified taxonomy*

Description

get the data of specified taxonomy

Usage

```
get_taxadf(obj, ...)

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

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

Arguments

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

Value

phyloseq class contained tax data.frame and sample information.

Author(s)

Shuangbin Xu

Examples

```
library(ggplot2)
data(test_otu_data)
phytax <- get_taxadf(test_otu_data, taxlevel=2)
phytax
head(phyloseq::otu_table(phytax))
phybar <- ggbartax(phytax) +
         xlab(NULL) + ylab("relative abundance (%)")
```

---

get_upset *generate the dataset for upset of UpSetR*

Description

generate the dataset for upset of UpSetR

Usage

```
get_upset(obj, ...)

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

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

Arguments

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

Value

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

Author(s)

Shuangbin Xu

Examples

```
data(test_otu_data)
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")
```

get_varct.pcoa *get the contribution of variables*

Description

get the contribution of variables

Usage

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

get_varct(obj, ...)

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

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

Arguments

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

Value

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

Examples

```
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)
```

get_vennlist *generate a vennlist for VennDiagram*

Description

generate a vennlist for VennDiagram

Usage

```
get_vennlist(obj, ...)

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

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

Arguments

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

Value

return a list for VennDiagram.

Author(s)

Shuangbin Xu

Examples

```
data(test_otu_data)
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")
```

ggbartax *taxonomy barplot*

Description

taxonomy barplot

Usage

```
ggbartax(obj, ...)

ggbartaxa(obj, ...)

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

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

Arguments

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

`...` additional parameters, see `ggplot`

`mapping` set of aesthetic mapping of ggplot2, default is NULL, if the data is the data.frame for geom_bar, the mapping should be set.

`position` character, default is ‘stack‘.

`stat` character, default is ‘identity‘.

`width` numeric, the width of bar, default is 0.7.

`topn` integer, the top number of abundance taxonomy(feature).

`count` logical, whether show the relative abundance.

`sampleda` data.frame, (nrow sample * ncol factor), the sample information, if the data doesn’t contain the information.

`factorLevels` vector or list, the levels of the factors (contained names e.g. list(group=c("B","A","C")) or c(group=c("B","A","C"))), adjust the order of facet, default is NULL, if you want to order the levels of factor, you can set this.

`sampleLevels` vector, adjust the order of x axis e.g. c("sample2", "sample4", "sample3"), default is NULL.

`facetNames` character, default is NULL.

`plotgroup` logical, whether calculate the mean or median etc for each group, default is FALSE.

`groupfun` character, how to calculate for feature in each group, the default is ‘mean‘, this will plot the mean of feature in each group.

Value

barplot of tax

Author(s)

Shuangbin Xu

Examples

```
library(ggplot2)
data(test_otu_data)
otubar <- ggbartax(test_otu_data) +
        xlab(NULL) + ylab("relative abundance(%)")
```

ggbox *A box or violin plot with significance test*

Description

A box or violin plot with significance test

Usage

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

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

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

Arguments

`obj`	object, alphasample or data.frame (row sample x column features).
`factorNames`	character, the names of factor contained in sampleda.
`...`	additional arguments, see also `stat_signif`.
`sampleda`	data.frame, sample information if obj is data.frame, the sampleda should be provided.
`indexNames`	character, the vector character, should be the names of features contained object.
`geom`	character, "boxplot" or "violin", default is "boxplot".
`factorLevels`	list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.
`compare`	logical, whether test the features among groups,default is TRUE.
`testmethod`	character, the method of test, default is ‘wilcox.test‘. see also `stat_signif`.
`signifmap`	logical, whether the pvalue are directly written a annotaion or asterisks are used instead, default is (pvalue) FALSE. see also `stat_signif`.

`p_textsize` numeric, the size of text of pvalue or asterisks, default is 2.

`step_increase` numeric, see also `stat_signif`, default is 0.1.

`boxwidth` numeric, the width of boxplot when the geom is 'violin', default is 0.2.

`facetnrow` integer, the nrow of facet, default is 1.

`controlgroup` character, the names of control group, if it was set, the other groups will compare to it, default is NULL.

`comparelist` list, the list of vector, default is NULL.

Value

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

Author(s)

Shuangbin Xu

Examples

```
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)
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")))
```

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 'clustplotClass'
ggclust(
  obj,
  layout = "rectangular",
  factorNames = NULL,
  factorLevels = NULL,
  pointsize = 2,
  fontsize = 2.6,
  hjust = -0.1,
  settheme = TRUE,
  ...
)
```

Arguments

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

Value

the figures of hierarchical cluster.

Author(s)

Shuangbin Xu

Examples

```
#don't run in examples
#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))
```

ggdiffbox *boxplot for the result of diff_analysis*

Description

boxplot for the result of diff_analysis

Usage

```
ggdiffbox(obj, ...)

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

Arguments

`obj`	object, diffAnalysisClass class.
`...,`	additional arguments.
`geom`	character, "boxplot" or "violin", default is "boxplot".
`box_notch`	logical, see also ‘notch‘ of `geom_boxplot`, default is TRUE.
`box_width`	numeric, the width of boxplot, default is 0.05

`dodge_width` numeric, the width of dodge of boxplot, default is 0.6.

`addLDA` logical, whether add the plot to visulize the result of LDA, default is TRUE.

`factorLevels` list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.

`featurelist` vector, the character vector, the sub feature of originalD in diffAnalysisClass,default is NULL.

`removeUnknown` logical, whether remove the unknown taxonomy, default is TRUE.

`colorlist` character, the color vector, default is NULL.

`l_xlabtext` character, the x axis text of left panel, default is NULL.

Value

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

Author(s)

Shuangbin Xu

Examples

```
data(kostic2012crc)
kostic2012crc
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")))
```

ggdiffclade *plot the clade tree with highlight*

Description

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

Usage

```
ggdiffclade(obj, ...)

## S3 method for class 'data.frame'
ggdiffclade(
  obj,
  nodedf,
  factorName,
  layout = "circular",
  linewd = 0.6,
  skpointsize = 0.8,
  alpha = 0.4,
  taxlevel = 6,
  cladetext = 2,
  factorLevels = NULL,
  setColors = TRUE,
  xlim = 12,
  reduce = FALSE,
  type = "species",
  ...
)

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

Arguments

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

`...,` additional parameters.

`nodedf` data.frame, contained the tax and the factor information and(or pvalue).

`factorName` character, the names of factor in nodedf.

`layout` character, the layout of ggtree, but only "rectangular", "radial", "slanted", "inward_circular" and "circular" in here, default is circular.

`linewd` numeric, the size of segment of ggtree, default is 0.6.

`skpointsize` numeric, the point size of skeleton of tree, default is 0.8 .

`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.

`factorLevels` list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.

`setColors` logical, whether set the color of clade, default is TRUE, or set FALSE,then use 'scale_fill_manual' setting.

`xlim` numeric, the x limits, only works for 'inward_circular' layout, default is 12.

`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".

`removeUnknown` logical, whether do not show unknown taxonomy, default is TRUE.

Value

figures of tax clade show the significant different feature.

Author(s)

Shuangbin Xu

Examples

```
data(kostic2012crc)
kostic2012crc
head(phyloseq::sample_data(kostic2012crc),3)
kostic2012crc <- phyloseq::rarefy_even_depth(kostic2012crc,
                         rngseed=1024)
table(phyloseq::sample_data(kostic2012crc)$DIAGNOSIS)
set.seed(1024)
diffres <- diff_analysis(kostic2012crc, classgroup="DIAGNOSIS",
                        mlfun="lda", filtermod="fdr",
                        firstcomfun = "kruskal.test",
                        firstalpha=0.05, strictmod=TRUE,
                        secondcomfun = "wilcox.test",
                        subclmin=3, subclwilc=TRUE,
                        secondalpha=0.01, ldascore=3)
#library(ggplot2)
#diffcladeplot <- ggdiffclade(diffres,alpha=0.3, linewd=0.2,
#                         skpointsize=0.4,
#                         taxlevel=3,
#                         setColors=FALSE) +
#        scale_fill_manual(values=c('#00AED7',
#                                   '#FD9347',
#                                   '#C1E168'))
```

---

ggdifftaxbar *significantly discriminative feature barplot*

Description

significantly discriminative feature barplot

Usage

```
ggdifftaxbar(obj, ...)

ggdiffbartaxa(obj, ...)

## S4 method for signature 'diffAnalysisClass'
ggdifftaxbar(
  obj,
  filepath = NULL,
  output = "biomarker_barplot",
  removeUnknown = TRUE,
  figwidth = 6,
  figheight = 3,
  ylabel = "relative abundance",
  format = "pdf",
  dpi = 300,
  ...
)

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

Arguments

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

Value

the figures of features show the distributions in samples.

Author(s)

Shuangbin Xu

Examples

```
data(kostic2012crc)
kostic2012crc
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)
# not run in example
#ggdifftaxbar(diffres, output="biomarker_barplot")
```

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

Description

visualization of effect size by the Linear Discriminant Analysis or randomForest

Usage

```
ggeffectsize(obj, ...)

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

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

Arguments

obj	object, diffAnalysisClass see diff_analysis, or data.frame, contained effect size and the group information.
...	additional arguments.

`factorName` character, the column name contained group information in data.frame.

`effectsizename` character, the column name contained effect size information.

`factorLevels` list, the levels of the factors, default is NULL, if you want to order the levels of factor, you can set this.

`linecolor` character, the color of horizontal error bars, default is grey50.

`linewidth` numeric, the width of horizontal error bars, default is 0.4.

`lineheight` numeric, the height of horizontal error bars, default is 0.2.

`pointsize` numeric, the size of points, default is 1.5.

`setFacet` logical, whether use facet to plot, default is TRUE.

`removeUnknown` logical, whether do not show unknown taxonomy, default is TRUE.

Value

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

Author(s)

Shuangbin Xu

Examples

```
data(kostic2012crc)
kostic2012crc
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())
```

ggordpoint *ordination plotter based on ggplot2.*

Description

ordination plotter based on ggplot2.

Usage

```
ggordpoint(obj, ...)

## Default S3 method:
ggordpoint(
  obj,
  pc = c(1, 2),
  mapping = NULL,
  sampleda = NULL,
  factorNames = NULL,
  factorLevels = NULL,
  poinsize = 2,
  linesize = 0.3,
  arrowsize = 1.5,
  arrowlinecolour = "grey",
  ellipse = FALSE,
  showsample = FALSE,
  ellipse_pro = 0.9,
  ellipse_alpha = 0.2,
  biplot = FALSE,
  topn = 5,
  settheme = TRUE,
  speciesannot = FALSE,
  fontsize = 2.5,
  labelfactor = NULL,
  stroke = 0.1,
  fontface = "bold.italic",
  fontfamily = "sans",
  textlinesize = 0.02,
  ...
)

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

Arguments

obj	prcomp class or pcasample class,
...	additional parameters, see geom_text_repel.
pc	integer vector, the component index.
mapping	set of aesthetic mapping of ggplot2, default is NULL when your want to set it by yourself, only alpha can be setted, and the first element of factorNames has

been setted to map fill, and the second element of factorNames has been setted to map starshape, you can use scale_starshape_manual 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.

biplot logical, whether plot the species, default is FALSE.

topn integer or vector, the number species have top important contribution, default is 5.

settheme logical, whether set the theme for the plot, default is TRUE.

speciesannot logical, whether plot the species, default is FALSE.

fontsize numeric, the size of text, default is 2.5.

labelfactor character, the factor want to be show in label, default is NULL.

stroke numeric, the line size of points, default is 0.1.

fontface character, the font face, default is "blod.italic".

fontfamily character, the font family, default is "sans".

textlinesize numeric, the segment size in geom_text_repel.

Value

point figures of PCA or PCoA.

Author(s)

Shuangbin Xu

Examples

```
#don't run in examples
#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)
```

ggrarecurve *Rarefaction alpha index*

Description

Rarefaction alpha index

Usage

```
ggrarecurve(obj, ...)

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

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

## S3 method for class 'rarecurve'
ggrarecurve(
  obj,
  indexNames = "Observe",
  linesize = 0.5,
  facetnrow = 1,
  shadow = TRUE,
  factorNames,
  se = FALSE,
  method = "lm",
  formula = y ~ log(x),
  ...
)
```

Arguments

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

Value

figure of rarefaction curves

Author(s)

Shuangbin Xu

Examples

```
data(test_otu_data)
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))
```

---

import_dada2 *Import function to load the feature table and taxanomy table of dada2*

Description

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

Usage

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

Arguments

seqtab	matrix, feature table, the output of removeBimeraDenovo.
taxatab	matrix, a taxonomic table, the output of assignTaxonomy, or the ouput of addSpecies.
reftree	phylo or character, the phylo class of tree, or the tree file.
sampleda	data.frame or character, the data.frame of sample information, or the file of sample information, nrow samples X ncol factors.
btree	logical, whether building the tree, default is FALSE.
...,	additional parameters, see also build_tree.

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")
ps <- import_dada2(seqtab=seqtab, taxatab=taxa,
                   sampleda=sampleda)
ps
```

import_qiime2   *Import function to load the output of qiime2.*

Description

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

Usage

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

Arguments

otuqza	character, the file contained otu table, the ouput of qiime2.
taxaqza	character, the file contained taxonomy, the ouput of qiime2, default is NULL.
mapfilename	character, the file contained sample information, the tsv format, default is NULL.
refseqqza	character, the file contained refrentent sequences, default is NULL.
treeqza	character, the file contained the tree file, default is NULL.
build_tree	logical, whether building the tree, when the rownames of feature table contains the sequence, default is FALSE.
parallel	logical, whether parsing the column of taxonomy multi-parallel, default is FALSE.
...,	additional parameters, see alosbuild_tree.

Value

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")
ps <- import_qiime2(otuqza=otuqzafile, taxaqza=taxaqzafile,
                    mapfilename=mapfile)
ps
```

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

Description

a container for performing two or more sample test.

Usage

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

Arguments

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

Value

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

Author(s)

Shuangbin Xu

Examples

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

`ordplotClass-class` *ordplotClass class*

Description

ordplotClass class

Slots

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

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

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

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

`pcasample-class` *pcasample class*

Description

pcasample class

Slots

`pca` prcomp or pcoa object

`sampleda` associated sample information

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

Description

the function was designed to read the ouput of qiime2.

Usage

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

Arguments

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

Value

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

Examples

```
otuqzafile <- system.file("extdata", "table.qza",
                          package="MicrobiotaProcess")
otuqza <- read_qza(otuqzafile)
str(otuqza)
```

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

Description

method extensions to show for diffAnalysisClass objects.

Usage

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

Arguments

object	object, ‘diffAnalysisClass‘ class

Value

print info

Author(s)

Shuangbin Xu

Examples

```
# don't run in examples
#data(kostic2012crc)
#kostic2012crc
#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)
```

split_data *Split Large Vector or DataFrame*

Description

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

Usage

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

Arguments

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

Value

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

Author(s)

Shuangbin Xu

Examples

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

---

split_str_to_list *split a dataframe contained one column*

Description

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

Usage

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

Arguments

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

Value

data.frame of strdataframe by sep.

Author(s)

Shuangbin Xu

Examples

```
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)
```

---

theme_taxbar *theme_taxbar*

Description

theme_taxbar

Usage

```
theme_taxbar(
  axis.text.x = element_text(angle = -45, hjust = 0, size = 12),
  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

```
library(ggplot2)
data(test_otu_data)
otubar <- ggbartax(test_otu_data, settheme=FALSE) +
    xlab(NULL) + ylab("relative abundance(%)") +
    theme_taxbar()
```

Index