Package 'file2meco'

November 16, 2021

Type Package

Title Transform Files to 'microtable' Object with 'microeco' Package

Version 0.2.0

Author Chi Liu [aut, cre]

Maintainer Chi Liu <liuchi0426@126.com>

Description Transform output files of some tools to the 'microtable' object of 'microtable' class in 'microeco' package. The 'microtable' class is the basic class in 'microeco' package and is necessary for the downstream microbial community data analysis.

Depends R (>= 3.5.0)

Imports R6, microeco, ape, magrittr, dplyr, tidyr, yaml, rhdf5, Matrix

Suggests Biostrings, seqinr, phyloseq

License GPL-3

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2021-11-16 05:40:07 UTC

RoxygenNote 7.1.1

R topics documented:

`check_match_table` *Replace the names use match table*

Description

Replace the names use match table

Usage

```
check_match_table(match_table = NULL, abund_new = NULL)
```

Arguments

`match_table`	default NULL; character or data.frame; matching table used.
`abund_new`	default NULL; data.frame; the abundance table used.

Value

new abundance table.

`check_sample_table` *Read sample table*

Description

Read sample table

Usage

```
check_sample_table(sample_data = NULL)
```

Arguments

`sample_data`	default NULL; character or data.frame; matching table used.

Value

sample information table.

CHOCOPhlAn_taxonomy *The CHOCOPhlAn_taxonomy data*

Description

The CHOCOPhlAn_taxonomy data is used for the parsing the 'HUMAaN' metagenomic results and add the taxonomy hierarchical information to the 'tax_table'.

Usage

```
data(CHOCOPhlAn_taxonomy)
```

humann2meco *Transform 'HUMAnN' metagenomic results to 'microtable' object.*

Description

Transform 'HUMAnN' metagenomic results to microtable object, reference: Franzosa et al. (2018) <doi:10.1038/s41592-018-0176-y>.

Usage

```
humann2meco(
  abund_table,
  db = c("MetaCyc", "KEGG")[1],
  sample_data = NULL,
  match_table = NULL
)
```

Arguments

abund_table	'HUMAnN' output abundance table, see the example.
db	default "MetaCyc"; either "MetaCyc" or "KEGG"; the pathway database used in the abund_table file generation.
sample_data	default NULL; the sample metadata table, must be tab or comma seperated file, generally, a file with suffix "tsv" or "csv"..
match_table	default NULL; a two column table used to replace the sample names in 'HUMAnN abundance result; Remember just two columns with no column names; The first column must be sample names used in abund_table, the second column is the new sample names, e.g. the rownames in sample_table. See the example files.

Value

microtable object.

Examples

```
library(file2meco)
library(microeco)
library(magrittr)
sample_file_path <- system.file("extdata", "example_metagenome_sample_info.tsv",
  package="file2meco")
match_file_path <- system.file("extdata", "example_metagenome_match_table.tsv", package="file2meco")
# MetaCyc pathway examples
# use the raw data files stored inside the package for MetaCyc pathway database based analysis
abund_file_path <- system.file("extdata", "example_HUMAnN_MetaCyc_abund.tsv", package="file2meco")
# the default db is "MetaCyc"
humann2meco(abund_table = abund_file_path, db = "MetaCyc")
humann2meco(abund_table = abund_file_path, db = "MetaCyc", sample_data = sample_file_path,
  match_table = match_file_path)
test <- humann2meco(abund_table = abund_file_path, db = "MetaCyc", sample_data = sample_file_path,
  match_table = match_file_path)
test$tidy_dataset()
# rel = FALSE donot use relative abundance
test$cal_abund(select_cols = 1:3, rel = FALSE)
test$taxa_abund$Superclass1 %<>% .[!grepl("unclass", rownames(.)), ]
test1 <- trans_abund$new(test, taxrank = "Superclass1", ntaxa = 10)
test1$plot_bar(facet = "Group", ylab_title = "Abundance (RPK)")
# select both function and taxa
test$cal_abund(select_cols = c("Superclass1", "Phylum", "Genus"), rel = TRUE)
test1 <- trans_abund$new(test, taxrank = "Phylum", ntaxa = 10, delete_part_prefix = TRUE)
test1$plot_bar(facet = "Group")
test$taxa_abund$Phylum %<>% .[!grepl("unclass", rownames(.)), ]
test1 <- trans_abund$new(test, taxrank = "Phylum", ntaxa = 10, delete_part_prefix = TRUE)
test1$plot_bar(facet = "Group")
# functional biomarker
test$cal_abund(select_cols = 1:3, rel = TRUE)
test$taxa_abund$Superclass1 %<>% .[!grepl("unclass", rownames(.)), ]
test$taxa_abund$Superclass2 %<>% .[!grepl("unclass", rownames(.)), ]
test$taxa_abund$pathway %<>% .[!grepl("unclass", rownames(.)), ]
test1 <- trans_diff$new(test, method = "lefse", group = "Group")
test1$plot_lefse_bar(use_number = 1:20)
# taxa biomarker
test$cal_abund(select_cols = 4:9, rel = TRUE)
test$taxa_abund$Phylum %<>% .[!grepl("unclass", rownames(.)), ]
test1 <- trans_diff$new(test, method = "lefse", group = "Group")
test1$plot_lefse_bar(LDA_score = 2)
############################################################
# KEGG pathway examples
abund_file_path <- system.file("extdata", "example_HUMAnN_KEGG_abund.tsv", package="file2meco")
humann2meco(abund_table = abund_file_path, db = "KEGG")
test <- humann2meco(abund_table = abund_file_path, db = "KEGG",
  sample_data = sample_file_path, match_table = match_file_path)
test$tax_table %<>% subset(level1 != "unclassified")
test$tidy_dataset()
# rel = FALSE donot use relative abundance
test$cal_abund(select_cols = 1:3, rel = FALSE)
```

```
test1 <- trans_abund$new(test, taxrank = "level2", ntaxa = 10)
test1$plot_bar(facet = "Group", ylab_title = "Abundance (RPK)")
# select both function and taxa
test$cal_abund(select_cols = c("level1", "Phylum", "Genus"), rel = TRUE)
test1 <- trans_abund$new(test, taxrank = "Phylum", ntaxa = 10, delete_part_prefix = TRUE)
test1$plot_bar(facet = "Group")
# functional biomarker
test$cal_abund(select_cols = 1:3, rel = TRUE)
test1 <- trans_diff$new(test, method = "lefse", group = "Group")
test1$plot_lefse_bar(LDA_score = 3)
# taxa biomarker
test$cal_abund(select_cols = 4:9, rel = TRUE)
test1 <- trans_diff$new(test, method = "lefse", group = "Group")
test1$plot_lefse_bar(LDA_score = 2)
```

`meco2phyloseq` *Transform 'microtable' object of 'microeco' package to the 'phyloseq' object of 'phyloseq' package.*

Description

Transform 'microtable' object of 'microeco' package to the 'phyloseq' object of 'phyloseq' package.

Usage

```
meco2phyloseq(dataset)
```

Arguments

`dataset`	a microtable object.

Value

phyloseq object.

Examples

```
library(microeco)
data("dataset")
meco2phyloseq(dataset)
```

MetaCyc_pathway_map *The MetaCyc_pathway_map data*

Description

The MetaCyc_pathway_map data is a manually curated 'MetaCyc' pathway hierarchical structure data. It is used for the parsing the 'HUMAaN' metagenomic abundance table associated with 'MetaCyc' database. Currently, only superclass 1, 2 and the pathway are used in this data.

Usage

```
data(MetaCyc_pathway_map)
```

metacyc_pathway_website *Get the website for a 'MetaCyc' pathway name*

Description

Get the website for a 'MetaCyc' pathway name

Usage

```
metacyc_pathway_website(pathway = NULL)
```

Arguments

pathway	default NULL; character vector; one or more MetaCyc pathway names.

Value

character vector.

Examples

```
metacyc_pathway_website("FOLSYN-PWY")
```

mpa2meco *Transform metagenomic classification results of 'mpa' format to 'microtable' object.*

Description

Transform the classification results of mpa (MetaPhlAn) format to microtable object, such as MetaPhlAn and Kraken2 results. Kraken2 results can be obtained by merge_metaphlan_tables.py from MetaPhlAn or combine_mpa.py from KrakenTools (https://ccb.jhu.edu/software/krakentools/).

Usage

```
mpa2meco(abund_table, sample_data = NULL, match_table = NULL)
```

Arguments

abund_table	'mpa' format abundance table, see the example.
sample_data	default NULL; the sample metadata table, must be tab or comma seperated file, generally, a file with suffix "tsv" or "csv"..
match_table	default NULL; a two column table used to replace the sample names in 'HUMAnN abundance result; Remember just two columns with no column names; The first column must be sample names used in abund_table, the second column is the new sample names, e.g. the rownames in sample_table. See the example files.

Value

microtable object.

Examples

```
# use the raw data files stored inside the package
abund_file_path <- system.file("extdata", "example_kraken2_merge.txt", package="file2meco")
sample_file_path <- system.file("extdata", "example_metagenome_sample_info.tsv",
  package="file2meco")
match_file_path <- system.file("extdata", "example_metagenome_match_table.tsv", package="file2meco")
library(microeco)
library(file2meco)
library(magrittr)
mpa2meco(abund_table = abund_file_path)
test <- mpa2meco(abund_table = abund_file_path, sample_data = sample_file_path,
  match_table = match_file_path)
# make the taxonomy standard for the following analysis
test$tax_table %<>% tidy_taxonomy
test$tidy_dataset()
```

ncyc2meco *Transform 'Ncyc' metagenomic abundance to 'microtable' object.*

Description

Transform 'Ncyc' metagenomic abundance to microtable object. Reference: Qichao et al. (2019) <doi: 10.1093/bioinformatics/bty741>.

Usage

```
ncyc2meco(abund_table, sample_data = NULL, match_table = NULL)
```

Arguments

abund_table	'Ncyc' software output abundance table, see the example file.
sample_data	default NULL; the sample metadata table; data.frame or character for the path; A file path must be tab or comma seperated file, generally, a file with suffix "tsv" or "csv".
match_table	default NULL; data.frame or character for the path; should be two column table used to replace the sample names in abundance result; Remember just two columns with no column names; The first column must be sample names used in abund_table, the second column is the new sample names, e.g. the rownames in sample_table. See the example files; A file path must be tab or comma seperated file, e.g. a file with suffix "tsv" or "csv".

Value

microtable object.

Examples

```
# use the raw data files stored inside the package
abund_file_path <- system.file("extdata", "example_Ncyc_table.tsv", package="file2meco")
sample_file_path <- system.file("extdata", "example_metagenome_sample_info.tsv",
  package="file2meco")
match_file_path <- system.file("extdata", "example_metagenome_match_table.tsv", package="file2meco")
library(microeco)
library(file2meco)
library(magrittr)
ncyc2meco(abund_table = abund_file_path)
test <- ncyc2meco(abund_table = abund_file_path, sample_data = sample_file_path,
  match_table = match_file_path)
test$tidy_dataset()
# use split_group = TRUE to calculate the pathway abundance with multipe map correspondance
test$cal_abund(select_cols = 1:2, rel = TRUE, split_group = TRUE, split_column = "Pathway")
test$taxa_abund$Pathway %<>% .[!grepl("unclass", rownames(.)), ]
test1 <- trans_abund$new(test, taxrank = "Pathway")
```

```
test1$plot_bar(bar_type = "notfull")
# for gene abundance, no splitting on the Pathway
test$cal_abund(select_cols = 1:2, rel = TRUE, split_group = FALSE)
test$taxa_abund$Gene %<>% .[!grepl("unclass", rownames(.)), ]
test1 <- trans_abund$new(test, taxrank = "Gene")
test1$plot_bar(bar_type = "notfull")
```

---

ncyc_map *The ncyc_map data*

Description

The ncyc_map data is used for the parsing the 'Ncyc' metagenomic results and add the N cycle pathway information to the 'tax_table' of 'microtable' object.

Usage

```
data(ncyc_map)
```

---

phyloseq2meco *Transform the 'phyloseq' object of 'phyloseq' package to 'microtable' object of 'microeco' package.*

Description

Transform the 'phyloseq' object of 'phyloseq' package to 'microtable' object of 'microeco' package.

Usage

```
phyloseq2meco(physeq)
```

Arguments

physeq	a phyloseq object.

Value

microtable object.

Examples

```
library(phyloseq)
data("GlobalPatterns")
phyloseq2meco(GlobalPatterns)
```

qiime1meco *Transform 'QIIME' results to 'microtable' object.*

Description

Transform 'QIIME' results to microtable object.

Usage

```
qiime1meco(
  otu_table,
  commented = TRUE,
  sample_data = NULL,
  phylo_tree = NULL,
  rep_fasta = NULL
)
```

Arguments

otu_table	the otu table generated from 'QIIME'. Taxonomic information should be in the end of the file.
commented	default TRUE; whether there is a commented first line in the otu_table.
sample_data	default NULL; If provided, must be tab or comma seperated file, generally, a file with suffix "tsv" or "csv".
phylo_tree	default NULL; the phylogenetic tree; generally, a file with suffix "tre".
rep_fasta	default NULL; the representative sequences; a fasta file, generally with suffix "fasta" or "fna" or "fa".

Value

microtable object.

Examples

```
## Not run:
# use the raw data files stored inside the package
otu_file_path <- system.file("extdata", "otu_table_raw.txt", package="file2meco")
sample_file_path <- system.file("extdata", "sample_info.csv", package="file2meco")
phylo_file_path <- system.file("extdata", "rep_phylo.tre", package="file2meco")
rep_fasta_path <- system.file("extdata", "rep.fna", package="file2meco")
qiime1meco(otu_table = otu_file_path, commented = FALSE, sample_data = sample_file_path)
qiime1meco(otu_table = otu_file_path, commented = FALSE, sample_data = sample_file_path,
  phylo_tree = phylo_file_path)
qiime1meco(otu_table = otu_file_path, commented = FALSE, sample_data = sample_file_path,
  phylo_tree = phylo_file_path, rep_fasta = rep_fasta_path)

## End(Not run)
```

qiime2meco *Transform 'QIIME2' results to 'microtable' object.*

Description

Transform 'QIIME2' qza results to microtable object.

Usage

```
qiime2meco(
  ASV_data,
  sample_data = NULL,
  taxonomy_data = NULL,
  phylo_tree = NULL,
  rep_fasta = NULL
)
```

Arguments

ASV_data	the ASV data, such as the data2_table.qza.
sample_data	default NULL; the sample metadata table, such as the sample-metadata.tsv.
taxonomy_data	default NULL; the taxonomy data, such as the taxonomy.qza.
phylo_tree	default NULL; the phylogenetic tree, such as the tree.qza.
rep_fasta	default NULL; the representative sequences, such as the dada2_rep_set.qza.

Value

microtable object.

Examples

```
## Not run:
# The data files is downloaded from https://docs.qiime2.org/2020.8/tutorials/pd-mice/
#   and stored inside the package.
abund_file_path <- system.file("extdata", "dada2_table.qza", package="file2meco")
sample_file_path <- system.file("extdata", "sample-metadata.tsv", package="file2meco")
taxonomy_file_path <- system.file("extdata", "taxonomy.qza", package="file2meco")
phylo_file_path <- system.file("extdata", "tree.qza", package="file2meco")
rep_fasta_path <- system.file("extdata", "dada2_rep_set.qza", package="file2meco")
qiime2meco(ASV_data = abund_file_path)
qiime2meco(ASV_data = abund_file_path, sample_data = sample_file_path)
qiime2meco(ASV_data = abund_file_path, sample_data = sample_file_path,
  taxonomy_data = taxonomy_file_path)
qiime2meco(ASV_data = abund_file_path, sample_data = sample_file_path,
  taxonomy_data = taxonomy_file_path, phylo_tree = phylo_file_path)
qiime2meco(ASV_data = abund_file_path, sample_data = sample_file_path, taxonomy_data =
  taxonomy_file_path, phylo_tree = phylo_file_path, rep_fasta = rep_fasta_path)

## End(Not run)
```

Index