Package ‘ggpicrust2’

June 9, 2023

Type Package

Title Make 'PICRUSt2' Output Analysis and Visualization Easier

Version 1.7.1

Description Provides a convenient way to analyze and visualize 'PICRUSt2' output with pre-defined plots and functions. Allows for generating statistical plots about microbiome functional predictions and offers customization options. Features a one-click option for creating publication-level plots, saving time and effort in producing professional-grade figures. Streamlines the 'PICRUSt2' analysis and visualization process. For more details, see Yang et al. (2023) <arXiv:2303.10388>.

BugReports https://github.com/cafferychen777/ggpicrust2/issues

URL https://github.com/cafferychen777/ggpicrust2

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports ALDEx2, aplot, DESeq2, dplyr, edgeR, GGally, ggplot2, grid, ggh4x, lefser, limma, Maaslin2, metagenomeSeq, MicrobiomeStat, readr, stats, SummarizedExperiment, tibble, tidyr, ggprism, patchwork, circlize

Depends R (>= 3.5.0)

Suggests covr, testthat (>= 3.0.0), cowplot, Biobase, devtools, KEGGREST, ComplexHeatmap, ggforce, ggplotify, BiocGenerics, BiocManager, magrittr, utils, knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Chen Yang [aut, cre], Liangliang Zhang [aut]

Maintainer Chen Yang <cafferychen7850@gmail.com>

Repository CRAN

Date/Publication 2023-06-09 16:20:10 UTC
R topics documented:

- compare_daa_results
- compare_metagenome_results
- daa_annotated_results_df
- daa_results_df
- ggpicrust2
- import_MicrobiomeAnalyst_daa_results
- kegg_abundance
- ko2kegg_abundance
- ko_abundance
- metacyc_abundance
- metadata
- pathway_annotation
- pathway_daa
- pathway_errorbar
- pathway_heatmap
- pathway_pca

Index

**compare_daa_results**  
*Compare the Consistency of Statistically Significant Features*

Description

This function compares the consistency and inconsistency of statistically significant features obtained using different methods in `pathway_daa` from the `ggpicrust2` package. It creates a report showing the number of common and different features identified by each method, and the features themselves.

Arguments

- **daa_results_list**  
  A list of data frames containing statistically significant features obtained using different methods.
- **method_names**  
  A character vector of names for each method used.
- **p_values_threshold**  
  A numeric value representing the threshold for the p-values. Features with p-values less than this threshold are considered statistically significant. Default is 0.05.

Value

A data frame with the comparison results. The data frame has the following columns:

- method: The name of the method.
- num_features: The total number of statistically significant features obtained by the method.
• **num_common_features**: The number of features that are common to other methods.
• **num_diff_features**: The number of features that are different from other methods.
• **diff_features**: The names of the features that are different from other methods.

**Examples**

```r
library(magrittr)
library(ggpicrust2)
library(tibble)
data("metacyc_abundance")
data("metadata")

# Run pathway_daa function for multiple methods
methods <- c("ALDEx2", "DESeq2", "edgeR")
daas_results_list <- lapply(methods, function(method) {
  pathway_daa(abundance = metacyc_abundance %>% column_to_rownames("pathway"),
              metadata = metadata, group = "Environment", daa_method = method)
})

# Compare results across different methods
comparison_results <- compare_daa_results(daa_results_list = daas_results_list,
                                           method_names = c("ALDEx2_Welch's t test", "ALDEx2_Wilcoxon rank test", "DESeq2", "edgeR"))
```

**Description**

**Compare Metagenome Results**

**Arguments**

- **metagenomes**: A list of metagenomes matrices with rows as KOs and columns as samples. Each matrix in the list should correspond to a different metagenome.
- **names**: A vector of names for the metagenomes in the same order as in the `metagenomes` list.
- **daa_method**: A character specifying the method for differential abundance analysis (DAA). Possible choices are: "ALDEx2", "DESeq2", "edgeR", "limma voom", "metagenomeSeq", "LinDA", "Maaslin2", and "Lefse". The default is "ALDEx2".
- **p.adjust**: A character specifying the method for p-value adjustment. Possible choices are: "BH" (Benjamini-Hochberg), "holm", "bonferroni", "hochberg", "fdr", and "none". The default is "BH".
- **reference**: A character specifying the reference group level for DAA. This parameter is used when there are more than two groups. The default is NULL.
Value

A list containing two elements:

- "daa": a list of results from the 'pathway_daa' function. Each result is a data frame containing the differential abundance analysis results with columns for the feature ID, the test statistic, the raw p-value, and the adjusted p-value.
- "correlation": a list with two elements: "cor_matrix" and "p_matrix", which are matrices of Spearman correlation coefficients and their corresponding p-values, respectively, between every pair of metagenomes.

Examples

```r
library(dplyr)
library(ComplexHeatmap)

# Generate example data
set.seed(123)

# First metagenome
metagenome1 <- abs(matrix(rnorm(1000), nrow = 100, ncol = 10))
rownames(metagenome1) <- paste0("KO", 1:100)
colnames(metagenome1) <- paste0("sample", 1:10)

# Second metagenome
metagenome2 <- abs(matrix(rnorm(1000), nrow = 100, ncol = 10))
rownames(metagenome2) <- paste0("KO", 1:100)
colnames(metagenome2) <- paste0("sample", 1:10)

# Put the metagenomes into a list
metagenomes <- list(metagenome1, metagenome2)

# Define names
names <- c("metagenome1", "metagenome2")

# Call the function
results <- compare_metagenome_results(metagenomes, names)

# Print the correlation matrix
print(results$correlation$cor_matrix)

# Print the p-value matrix
print(results$correlation$p_matrix)
```

daa_annotated_results_df

Differentially Abundant Analysis Results with Annotation

Description

This is a result dataset after processing 'kegg_abundance' through the 'pathway_daa' with the LinDA method and further annotation with 'pathway_annotation'.

Usage

`daa_annotated_results_df`
**Format**

A data frame with 10 variables:

- **adj_method**  Method used for adjusting p-values.
- **feature**  Feature being tested.
- **group1**  One group in the comparison.
- **group2**  The other group in the comparison.
- **method**  Statistical test used.
- **p_adjust**  Adjusted p-value.
- **p_values**  P-values from the statistical test.
- **pathway_class**  Class of the pathway.
- **pathway_description**  Description of the pathway.
- **pathway_map**  Map of the pathway.
- **pathway_name**  Name of the pathway.

**Source**

From ggpicrust2 package demonstration.

**References**

Format

A data frame with columns:

- **adj_method**  Method used for p-value adjustment.
- **feature**   The feature (pathway) being compared.
- **group1**     The first group in the comparison.
- **group2**     The second group in the comparison.
- **method**     The method used for the comparison.
- **p_adjust**   The adjusted p-value from the comparison.
- **p_values**   The raw p-value from the comparison.

Source

From ggpicrust2 package demonstration.

References


---

**ggpicrust2**

This function integrates pathway name/description annotations, ten of the most advanced differential abundance (DA) methods, and visualization of DA results.

Description

This function integrates pathway name/description annotations, ten of the most advanced differential abundance (DA) methods, and visualization of DA results.

Usage

```r
ggpicrust2(
    file = NULL,
    data = NULL,
    metadata,
    group,
    pathway,
    daa_method = "ALDEx2",
    ko_to_kegg = FALSE,
    p.adjust = "BH",
    order = "group",
    p_values_bar = TRUE,
    x_lab = NULL,
    select = NULL,
    reference = NULL,
    colors = NULL
)
```
Arguments

- **file**: A character string representing the file path of the input file containing KO abundance data in `picrust2` export format. The input file should have KO identifiers in the first column and sample identifiers in the first row. The remaining cells should contain the abundance values for each KO-sample pair.

- **data**: An optional data.frame containing KO abundance data in the same format as the input file. If provided, the function will use this data instead of reading from the file. By default, this parameter is set to NULL.

- **metadata**: A tibble, consisting of sample information

- **group**: A character, name of the group

- **pathway**: A character, consisting of "EC", "KO", "MetaCyc"

- **daa_method**: a character specifying the method for differential abundance analysis, choices are: - "ALDEX2": ANOVA-Like Differential Expression tool for high throughput sequencing data - "DESeq2": Differential expression analysis based on the negative binomial distribution using DESeq2 - "edgeR": Exact test for differences between two groups of negative-binomially distributed counts using edgeR - "limma voom": Limma-voom framework for the analysis of RNA-seq data - "metagenomeSeq": Fit logistic regression models to test for differential abundance between groups using metagenomeSeq - "LinDA": Linear models for differential abundance analysis of microbiome compositional data - "Maaslin2": Multivariate Association with Linear Models (MaAsLin2) for differential abundance analysis

- **ko_to_kegg**: A character to control the conversion of KO abundance to KEGG abundance

- **p.adjust**: a character specifying the method for p-value adjustment, choices are: - "BH": Benjamini-Hochberg correction - "holm": Holm’s correction - "bonferroni": Bonferroni correction - "hochberg": Hochberg’s correction - "fdr": False discovery rate correction - "none": No p-value adjustment.

- **order**: A character to control the order of the main plot rows

- **p_values_bar**: A character to control if the main plot has the p_values bar

- **x_lab**: A character to control the x-axis label name, you can choose from "feature","pathway_name" and "description"

- **select**: A vector consisting of pathway names to be selected

- **reference**: A character, a reference group level for several DA methods

- **colors**: A vector consisting of colors number

Value

da.a.results.df, a dataframe of DA results

Examples

```
## Not run:
# Load necessary data: abundance data and metadata
abundance_file <- "path/to/your/abundance_file.tsv"
metadata <- read.csv("path/to/your/metadata.csv")
```
# Run ggpicrust2 with input file path
results_file_input <- ggpicrust2(file = abundance_file,
metadata = metadata,
group = "your_group_column",
pathway = "KO",
da_method = "LinDA",
ko_to_kegg = "TRUE",
order = "pathway_class",
p_values_bar = TRUE,
x_lab = "pathway_name")

# Run ggpicrust2 with imported data.frame
abundance_data <- read_delim(abundance_file, delim="\t", col_names=TRUE, trim_ws=TRUE)

# Run ggpicrust2 with input data
results_data_input <- ggpicrust2(data = abundance_data,
metadata = metadata,
group = "your_group_column",
pathway = "KO",
da_method = "LinDA",
ko_to_kegg = "TRUE",
order = "pathway_class",
p_values_bar = TRUE,
x_lab = "pathway_name")

# Access the plot and results dataframe for the first DA method
example_plot <- results_file_input[[1]]$plot
example_results <- results_file_input[[1]]$results

# Use the example data in ggpicrust2 package
data(ko_abundance)
data(metadata)
results_file_input <- ggpicrust2(data = ko_abundance,
metadata = metadata,
group = "Environment",
pathway = "KO",
da_method = "LinDA",
ko_to_kegg = "TRUE",
order = "pathway_class",
p_values_bar = TRUE,
x_lab = "pathway_name")

# Analyze the EC or MetaCyc pathway
data(metacyc_abundance)
results_file_input <- ggpicrust2(data = metacyc_abundance,
metadata = metadata,
group = "Environment",
pathway = "MetaCyc",
da_method = "LinDA",
ko_to_kegg = FALSE,
order = "group",
p_values_bar = TRUE,
x_lab = "description")
import_MicrobiomeAnalyst_daa_results

Import Differential Abundance Analysis (DAA) results from MicrobiomeAnalyst

Description

This function imports DAA results from an external platform such as MicrobiomeAnalyst. It can be used to compare the results obtained from different platforms.

Arguments

- **file_path**
  - a character string specifying the path to the CSV file containing the DAA results from MicrobiomeAnalyst. If this parameter is NULL and no data frame is provided, an error will be thrown. Default is NULL.

- **data**
  - a data frame containing the DAA results from MicrobiomeAnalyst. If this parameter is NULL and no file path is provided, an error will be thrown. Default is NULL.

- **method**
  - a character string specifying the method used for the DAA. This will be added as a new column in the returned data frame. Default is "MicrobiomeAnalyst".

- **group_levels**
  - a character vector specifying the group levels for the DAA. This will be added as new columns in the returned data frame. Default is c("control", "treatment").

Value

- a data frame containing the DAA results from MicrobiomeAnalyst with additional columns for the method and group levels.

Examples

```r
## Not run:
# Assuming you have a CSV file named "DAA_results.csv" in your current directory
daar_results <- import_MicrobiomeAnalyst_daa_results(file_path = "DAA_results.csv")
## End(Not run)
```
ko2kegg_abundance

Description
A dataset derived from 'ko_abundance' by the function 'ko2kegg_abundance' in the ggpicrust2 package. Each row corresponds to a KEGG pathway, and each column corresponds to a sample.

Usage
ko2kegg_abundance

Format

Source
From ggpicrust2 package demonstration.

References

ko2kegg_abundance

Convert KO abundance in picrust2 export files to KEGG pathway abundance

Description
This function takes a file containing KO (KEGG Orthology) abundance data in picrust2 export format and converts it to KEGG pathway abundance data. The input file should be in .tsv, .txt, or .csv format.

Usage
ko2kegg_abundance(file = NULL, data = NULL)
Arguments

file  A character string representing the file path of the input file containing KO abundance data in picrust2 export format. The input file should have KO identifiers in the first column and sample identifiers in the first row. The remaining cells should contain the abundance values for each KO-sample pair.

data  An optional data.frame containing KO abundance data in the same format as the input file. If provided, the function will use this data instead of reading from the file. By default, this parameter is set to NULL.

Value

A data frame with KEGG pathway abundance values. Rows represent KEGG pathways, identified by their KEGG pathway IDs. Columns represent samples, identified by their sample IDs from the input file. Each cell contains the abundance of a specific KEGG pathway in a given sample, calculated by summing the abundances of the corresponding KOs in the input file.

Examples

```r
## Not run:
library(ggpicrust2)
library(readr)

# Prepare an input file path
input_file <- "path/to/your/picrust2/results/pred_metagenome_unstrat.tsv"

# Run ko2kegg_abundance function
kegg_abundance <- ko2kegg_abundance(file = input_file)

# Alternatively, read the data from a file and use the data argument
file_path <- "path/to/your/picrust2/results/pred_metagenome_unstrat.tsv"
ko_abundance <- read_delim(file_path, delim = "\t")
kegg_abundance <- ko2kegg_abundance(data = ko_abundance)

# Print the result
print(abundance)

## End(Not run)
```

ko_abundance  KO Abundance Dataset

Description

This is a demonstration dataset from the ggpicrust2 package, representing the output of PICRUST2. Each row represents a KO (KEGG Orthology) group, and each column corresponds to a sample.

Usage

ko_abundance
Format


Source

From ggpicrust2 package demonstration.

References

Source

From ggpicrust2 package demonstration.

References


---

**Metadata for ggpicrust2 Demonstration**

**Description**

This is a demonstration dataset from the ggpicrust2 package. It provides the metadata required for the demonstration functions in the package. The dataset includes environmental information for each sample.

**Usage**

metadata

**Format**

A tibble with each row representing metadata for a sample.

**Sample1** Metadata for Sample1, including Environment

**Sample2** Metadata for Sample2, including Environment

... ...

**Source**

ggpicrust2 package demonstration.

**References**

**pathway_annotation**

Pathway information annotation of "EC", "KO", "MetaCyc" pathway

**Description**

This function has two primary use cases: 1. Annotating pathway information using the output file from PICRUSt2. 2. Annotating pathway information from the output of `pathway_daa` function, and converting KO abundance to KEGG pathway abundance when `ko_to_kegg` is set to TRUE.

**Usage**

```r
pathway_annotation(
  file = NULL,
  pathway = NULL,
  daa_results_df = NULL,
  ko_to_kegg = FALSE
)
```

**Arguments**

- **file**: A character, address to store PICRUSt2 export files. Provide this parameter when using the function for the first use case.
- **pathway**: A character, consisting of "EC", "KO", "MetaCyc"
- **daa_results_df**: A data frame, output of `pathway_daa`. Provide this parameter when using the function for the second use case.
- **ko_to_kegg**: A logical, decide if convert KO abundance to KEGG pathway abundance. Default is FALSE. Set to TRUE when using the function for the second use case.

**Value**

A data frame containing pathway annotation information. The data frame has the following columns:

- **feature**: The feature ID of the pathway (e.g., KO, EC, or MetaCyc ID).
- **description**: The description or name of the pathway.
- **Other columns depending on the input parameters and type of pathway.**

If `ko_to_kegg` is set to TRUE, the output data frame will also include the following columns:

- **pathway_name**: The name of the KEGG pathway.
- **pathway_description**: The description of the KEGG pathway.
- **pathway_class**: The class of the KEGG pathway.
- **pathway_map**: The KEGG pathway map ID.
Examples

```r
## Not run:
# Prepare the required input files and data frames
# Then, you can use the pathway_annotation function as follows:

# Use case 1: Annotating pathway information using the output file from PICRUSt2
result1 <- pathway_annotation(file = "path/to/picrust2/export/file.txt",
pathway = "KO",
daa_results_df = NULL,
ko_to_kegg = FALSE)

# Use case 2: Annotating pathway information from the output of pathway_daa function
# and converting KO abundance to KEGG pathway abundance
result2 <- pathway_annotation(file = NULL,
pathway = "KO",
daa_results_df = your_daa_results_df,
ko_to_kegg = TRUE)

## End(Not run)
```

---

### pathway_daa

Predictional functional pathway differential abundance (DA)

#### Description

Predictional functional pathway differential abundance (DA)

#### Usage

```r
pathway_daa(
  abundance,
  metadata,
  group,
  daa_method = "ALDEx2",
  select = NULL,
  p.adjust = "BH",
  reference = NULL
)
```

#### Arguments

- **abundance**: a data frame containing predicted functional pathway abundance, with pathways/features as rows and samples as columns. The column names of abundance should match the sample names in metadata. Pathway abundance values should be counts.
- **metadata**: a tibble containing samples information.
- **group**: a character specifying the group name for differential abundance analysis.
pathway_daa

daa_method  a character specifying the method for differential abundance analysis, choices are: - "ALDE2": ANOVA-Like Differential Expression tool for high throughput sequencing data - "DESeq2": Differential expression analysis based on the negative binomial distribution using DESeq2 - "edgeR": Exact test for differences between two groups of negative-binomially distributed counts using edgeR - "limma voom": Limma-voom framework for the analysis of RNA-seq data - "metagenomeSeq": Fit logistic regression models to test for differential abundance between groups using metagenomeSeq - "LinDA": Linear models for differential abundance analysis of microbiome compositional data - "Maaslin2": Multivariate Association with Linear Models (MaAsLin2) for differential abundance analysis - "Lefse": Linear discriminant analysis (LDA) effect size algorithm for high-dimensional microbiome data

select  a vector containing sample names for analysis, if NULL all samples are included. This parameter can be used to specify which samples are included in the differential abundance analysis. Default is NULL.

p.adjust  a character specifying the method for p-value adjustment, choices are: - "BH": Benjamini-Hochberg correction - "holm": Holm’s correction - "bonferroni": Bonferroni correction - "hochberg": Hochberg’s correction - "fdr": False discovery rate correction - "none": No p-value adjustment.

reference  a character specifying the reference group level, required for several differential abundance analysis methods such as LinDA, limme voom and Maaslin2. This parameter is used to specify the reference group when there are more than two groups. Default is NULL.

Value  a data frame containing the differential abundance analysis results.

Examples

library(ggpicrust2)
library(MicrobiomeStat)
library(tibble)
library(magrittr)
abundance <- data.frame(sample1 = c(10, 20, 30),
                        sample2 = c(20, 30, 40),
                        sample3 = c(30, 40, 50),
                        row.names = c("pathway1", "pathway2", "pathway3"))

metadata <- tibble::tibble(sample = paste0("sample", 1:3),
                            group = c("control", "control", "treatment"))

#Run pathway_daa function
result <- pathway_daa(abundance = abundance, metadata = metadata, group = "group",
                        daa_method = "LinDA")

data(metacyc_abundance)
data(metadata)
daas_results_df <- pathway_daa(metacyc_abundance %>%
                                 )
The function `pathway_errorbar()` is used to visualize the results of functional pathway differential abundance analysis as error bar plots.

**Description**

The function `pathway_errorbar()` is used to visualize the results of functional pathway differential abundance analysis as error bar plots.

**Arguments**

- **abundance**: A data frame with row names representing pathways and column names representing samples. Each element represents the relative abundance of the corresponding pathway in the corresponding sample.

- **daa_results_df**: A data frame containing the results of the differential abundance analysis of the pathways, generated by the `pathway_daa()` function. `x_lab` should be a column name of `daa_results_df`.

- **Group**: A data frame or a vector that assigns each sample to a group. The groups are used to color the samples in the figure.

- **ko_to_kegg**: A logical parameter indicating whether there was a conversion that convert ko abundance to kegg abundance.

- **p_values_threshold**: A numeric parameter specifying the threshold for statistical significance of differential abundance. Pathways with p-values below this threshold will be considered significant.

- **order**: A parameter controlling the ordering of the rows in the figure. The options are: "p_values" (order by p-values), "name" (order by pathway name), "group" (order by the group with the highest mean relative abundance), or "pathway_class" (order by the pathway category).

- **select**: A vector of pathway names to be included in the figure. This can be used to limit the number of pathways displayed. If NULL, all pathways will be displayed.

- **p_value_bar**: A logical parameter indicating whether to display a bar showing the p-value threshold for significance. If TRUE, the bar will be displayed.

- **colors**: A vector of colors to be used to represent the groups in the figure. Each color corresponds to a group.

- **x_lab**: A character string to be used as the x-axis label in the figure. The default value is "description" for KOs'descriptions and "pathway_name" for KEGG pathway names.

**Value**

A figure showing the error bar plot of the differential abundance analysis results for the functional pathways.
## Not run:

# Example 1: Analyzing KEGG pathway abundance
```r
data(metadata)
```
```r
kegg_abundance <- ko2kegg_abundance("path/to/your/pred_metagenome_unstrat.tsv")
```
```r
data(kegg_abundance)
```
```r
# Please change group to "your_group_column" if you are not using example dataset
group <- "Environment"
```
```r
daa_results_df <- pathway_daa(
  abundance = kegg_abundance,
  metadata = metadata,
  group = group,
  daa_method = "ALDEx2",
  select = NULL,
  reference = NULL
)
```
```r
# Please check the unique(daa_results_df$method) and choose one
daasub_method_results_df <- daa_results_df[daa_results_df$method == "ALDEx2_Welch's t test", ]
```
```r
daa_annotated_sub_method_results_df <- pathway_annotation(
  pathway = "KO",
  daa_results_df = daa_sub_method_results_df,
  ko_to_kegg = TRUE
)
```
```r
# Please change Group to metadata$your_group_column if you are not using example dataset
Group <- metadata$Environment
```
```r
p <- pathway_errorbar(
  abundance = kegg_abundance,
  daa_results_df = daa_annotated_sub_method_results_df,
  Group = Group,
  p_values_threshold = 0.05,
  order = "pathway_class",
  select = daa_annotated_sub_method_results_df %>%
    arrange(p_adjust) %>%
    slice(1:20) %>%
```

---

**Examples**

```r
## Not run:

# Example 1: Analyzing KEGG pathway abundance
metadata <- read_delim("path/to/your/metadata.txt",
  delim = "\t",
  escape_double = FALSE,
  trim_ws = TRUE)
```
```r
# data(metadata)
```
```r
kegg_abundance <- ko2kegg_abundance("path/to/your/pred_metagenome_unstrat.tsv")
```
```r
# data(kegg_abundance)
```
```r
# Please change group to "your_group_column" if you are not using example dataset
group <- "Environment"
```
```r
daa_results_df <- pathway_daa(
  abundance = kegg_abundance,
  metadata = metadata,
  group = group,
  daa_method = "ALDEx2",
  select = NULL,
  reference = NULL
)
```
```r
# Please check the unique(daa_results_df$method) and choose one
daasub_method_results_df <- daa_results_df[daa_results_df$method == "ALDEx2_Welch's t test", ]
```
```r
daa_annotated_sub_method_results_df <- pathway_annotation(
  pathway = "KO",
  daa_results_df = daa_sub_method_results_df,
  ko_to_kegg = TRUE
)
```
```r
# Please change Group to metadata$your_group_column if you are not using example dataset
Group <- metadata$Environment
```
```r
p <- pathway_errorbar(
  abundance = kegg_abundance,
  daa_results_df = daa_annotated_sub_method_results_df,
  Group = Group,
  p_values_threshold = 0.05,
  order = "pathway_class",
  select = daa_annotated_sub_method_results_df %>%
    arrange(p_adjust) %>%
    slice(1:20) %>%
```
select("feature") %>% pull(),
ko_to_kegg = TRUE,
p_value_bar = TRUE,
colors = NULL,
x_lab = "pathway_name"
)

# Example 2: Analyzing EC, MetaCyc, KO without conversions
metadata <- read_delim(
  "path/to/your/metadata.txt",
  delim = "\t",
  escape_double = FALSE,
  trim_ws = TRUE
)
# data(metadata)
ko_abundance <- read.delim("path/to/your/metacyc_abundance.tsv")
# data(metacyc_abundance)
group <- "Environment"
daas_results_df <- pathway_daa(
  abundance = metacyc_abundance %>% column_to_rownames("pathway"),
  metadata = metadata,
  group = group,
  daa_method = "LinDA",
  select = NULL,
  reference = NULL
)

daas_annotated_results_df <- pathway_annotation(
  pathway = "MetaCyc",
  daa_results_df = daas_results_df,
  ko_to_kegg = FALSE
)

Group <- metadata$Environment

p <- pathway_errorbar(
  abundance = metacyc_abundance %>% column_to_rownames("pathway"),
  daa_results_df = daas_annotated_results_df,
  Group = Group,
  p_values_threshold = 0.05,
  order = "group",
  select = NULL,
  ko_to_kegg = FALSE,
  p_value_bar = TRUE,
  colors = NULL,
  x_lab = "description"
)
pathway_heatmap

Create pathway heatmap

Description

This function creates a heatmap of the predicted functional pathway abundance data. The function first makes the abundance data relative, then converts the abundance data to a long format and orders the samples based on the environment information. The heatmap is then created using the ‘ggplot2’ library. The color palette, appearance and the color bar of the heatmap can be customized using the ‘scale_fill_gradientn’, ‘theme’ and ‘guides’ functions respectively.

Arguments

- **abundance**: A matrix or data frame of pathway abundance data, where columns correspond to samples and rows correspond to pathways.
- **metadata**: A data frame of metadata, where each row corresponds to a sample and each column corresponds to a metadata variable.
- **group**: A character string specifying the column name in the metadata data frame that contains the group variable.

Value

A ggplot heatmap object. The output is a ggplot object representing the heatmap of the predicted functional pathway abundance data. The heatmap visualizes the z score of pathways in different samples.

Examples

```r
library(ggpicrust2)
library(ggh4x)
library(dplyr)
library(tidyr)
library(tibble)
library(magrittr)

# Create example functional pathway abundance data
kegg_abundance_example <- matrix(rnorm(30), nrow = 3, ncol = 10)
colnames(kegg_abundance_example) <- paste0("Sample", 1:10)
rownames(kegg_abundance_example) <- c("PathwayA", "PathwayB", "PathwayC")

# Create example metadata
# Please ensure the sample IDs in the metadata have the column name "sample_name"
metadata_example <- data.frame(sample_name = colnames(kegg_abundance_example),
                               group = factor(rep("Control", "Treatment"), each = 5))

# Create a heatmap
```
pathway_pca

Perform Principal Component Analysis (PCA) on functional pathway abundance data and create visualizations of the PCA results.

Description

Perform Principal Component Analysis (PCA) on functional pathway abundance data and create visualizations of the PCA results.

Usage

pathway_pca(abundance, metadata, group)

Arguments

- **abundance**: A data frame, predicted functional pathway abundance.
- **metadata**: A tibble, consisting of sample information.
- **group**: A character, group name.

Value

A ggplot object showing the PCA results.

Examples

```r
library(magrittr)
library(dplyr)
library(tibble)

# Create example functional pathway abundance data
kegg_abundance_example <- matrix(rnorm(30), nrow = 3, ncol = 10)
colnames(kegg_abundance_example) <- paste0("Sample", 1:10)
```
rownames(kegg_abundance_example) <- c("PathwayA", "PathwayB", "PathwayC")

# Create example metadata
# Please ensure the sample IDs in the metadata have the column name "sample_name"
metadata_example <- data.frame(sample_name = colnames(kegg_abundance_example),
                               group = factor(rep(c("Control", "Treatment"), each = 5)))

pca_plot <- pathway_pca(kegg_abundance_example, metadata_example, "group")
print(pca_plot)

data("metacyc_abundance")
data("metadata")
pathway_pca(metacyc_abundance %>% column_to_rownames("pathway"), metadata, "Environment")
Index

* datasets
  daa_annotated_results_df, 4
  daa_results_df, 5
  kegg_abundance, 10
  ko_abundance, 11
  metacyc_abundance, 12
  metadata, 13

  compare_daa_results, 2
  compare_metagenome_results, 3

  daa_annotated_results_df, 4
  daa_results_df, 5

  ggpicrust2, 6

  import_MicrobiomeAnalyst_daa_results, 9

  kegg_abundance, 10
  ko2kegg_abundance, 10
  ko_abundance, 11

  metacyc_abundance, 12
  metadata, 13

  pathway_annotation, 14
  pathway_daa, 15
  pathway_errorbar, 17
  pathway_heatmap, 20
  pathway_pca, 21