Package ‘omu’

August 2, 2018

Title A Metabolomics Analysis Tool for Intuitive Figures and Convenient Metadata Collection

Version 1.0.2

Description Facilitates the creation of intuitive figures to describe metabolomics data by utilizing Kyoto Encyclopedia of Genes and Genomes (KEGG) hierarchy data, and gathers functional orthology and gene data using the package 'KEGGREST' to access the 'KEGG' API.

Depends R (>= 3.3.0)

biocViews

Imports plyr, dplyr, stringr, KEGGREST, reshape2, ggfortify, ggplot2, magrittr, tidyr

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL https://github.com/connor-reid-tiffany/Omu,
    https://www.kegg.jp/kegg/rest/keggapi.html

BugReports https://github.com/connor-reid-tiffany/Omu/issues

NeedsCompilation no

Author Connor Tiffany [aut, cre]

Maintainer Connor Tiffany <crtiffany@ucdavis.edu>

Repository CRAN

Date/Publication 2018-08-02 12:40:03 UTC


### Assign hierarchy metadata

**Description**

Assigns hierarchy metadata to a metabolomics count matrix using identifier values. It can assign KEGG compound hierarchy, orthology hierarchy, or organism hierarchy data.

**Usage**

```r
assign_hierarchy(count_data, keep_unknowns, identifier)
```

**Arguments**

- `count_data`: a metabolomics count data frame with either a KEGG compound, orthology, or a gene identifier column
- `keep_unknowns`: a boolean of either TRUE or FALSE. TRUE keeps unannotated compounds, FALSE removes them
- `identifier`: a string that is either "KEGG" for metabolite, "KO_Number" for orthology, "Prokaryote" for organism, or "Eukaryote" for organism

**Examples**

```r
assign_hierarchy(count_data = c57_nos2KO_mouse_countDF, keep_unknowns = TRUE, identifier = "KEGG")
```
Description

A dataset containing metabolomics counts for an experiment done using c57b6J wild type and c57b6J nos2 knockout mice.

Usage

c57_nos2KO_mouse_countDF

Format

A data frame with 668 rows and 36 variables:

c57_nos2KO_mouse_metadata

Description

A meta data file for the c57b6J metabolomics count matrix.

Usage

c57_nos2KO_mouse_metadata

Format

A data frame with 29 rows and 4 variables:
count_fold_changes  

*Get counts for significant fold changes by metabolite class.*

**Description**

Takes an input data frame from the output of `omu_summary` and creates a data frame of counts for significantly changed metabolites by class hierarchy data.

**Usage**

```r
count_fold_changes(count_data, ..., column, sig_threshold, keep_unknowns)
```

**Arguments**

- `count_data`: Output dataframe from the `omu_summary` function
- `...`: Either a Class or Subclass column as a string, i.e. "Class"
- `column`: The same value entered for the `...` argument, i.e. `column = "Class"
- `sig_threshold`: Significance threshold for compounds that go towards the count, `sig_threshold = 0.05`
- `keep_unknowns`: TRUE or FALSE for whether to drop compounds that weren’t assigned hierarchy metadata

**Examples**

```r
c57_nos2K0_mouse_countdf <- assign_hierarchy(c57_nos2K0_mouse_countDF, TRUE, "KEGG")

t_test_df <- omu_summary(count_data = c57_nos2K0_mouse_countDF, metadata = c57_nos2K0_mouse_metadata, numerator = "Strep", denominator = "Mock", response_variable = "Metabolite", factor = "Treatment", log_transform = TRUE, p_adjust = "BH")

fold_change_counts <- count_fold_changes(count_data = t_test_df, "Class", column = "Class", sig_threshold = 0.05, keep_unknowns = "FALSE")
```

---

**KEGG_gather**  

*Gather metadata from KEGG for metabolites*

**Description**

Method for gathering metadata from the KEGG API.
Usage

KEGG_gather(count_data)

## S3 method for class 'cpd'
KEGG_gather(count_data)

## S3 method for class 'rxn'
KEGG_gather(count_data)

## S3 method for class 'KO'
KEGG_gather(count_data)

Arguments

count_data A metabolomics count dataframe with a KEGG identifier columns

Examples

count_data <- assign_hierarchy(count_data = c57_nos2K0_mouse_countDF,
keep_unknowns = TRUE, identifier = "KEGG")

count_data <- subset(count_data, Subclass_2 == "Aldoses")

count_data <- KEGG_gather(count_data = count_data)
**omu_anova**

Performs an anova across all response variables. The function can take a maximum of 2 independent variables and perform an interaction term between them.

### Description

Perform anova Performs an anova across all response variables. The function can take a maximum of 2 independent variables and perform an interaction term between them.

### Usage

```r
omu_anova(count_data, metadata, response_variable, var1, var2, interaction, log_transform, p_adjust)
```

### Arguments

- `count_data`: A metabolomics count data frame
- `metadata`: Metadata dataframe for the metabolomics count data frame
- `response_variable`: String of the column header for the response variables, usually "Metabolite"
- `var1`: String of the first independent variable you wish to test
- `var2`: String of the second independent variable you wish to test. Optional parameter
- `interaction`: Boolean of TRUE or FALSE for whether or not you wish to model an interaction between independent variables. Optional parameter
- `log_transform`: Boolean of TRUE or FALSE for whether or not you wish to log transform your metabolite counts
- `p_adjust`: Method for p value adjustment, i.e. "BH"

### Examples

```r
anova_df <- omu_anova(count_data = c57_nos2K0_mouse_countDF, metadata = c57_nos2K0_mouse_metadata, response_variable = "Metabolite", var1 = "Treatment", var2 = "Background", log_transform = TRUE, p_adjust = "BH", interaction = TRUE)
```
**omu_summary**  

Performs t test, standard deviation, standard error, FDR correction, fold change, log2FoldChange. The order effects the fold change values.

**Description**

omu_summary Performs t test, standard deviation, standard error, FDR correction, fold change, log2FoldChange. The order effects the fold change values.

**Usage**

```r
omu_summary(count_data, metadata, numerator, denominator, response_variable, Factor, log_transform, p_adjust)
```

**Arguments**

- `count_data`: should be a metabolomics count data frame
- `metadata`: is meta data
- `numerator`: is the variable you wish to compare against the denominator, in quotes
- `denominator`: see above, in quotes
- `response_variable`: the name of the column with your response variables
- `Factor`: the column name for your independent variables
- `log_transform`: TRUE or FALSE value for whether or not log transformation of data is performed before the t test
- `p_adjust`: Method for adjusting the p value, i.e. "BH"

**Examples**

```r
omu_summary(count_data = c57_nos2KO_mouse_countDF, metadata = c57_nos2KO_mouse_metadata, numerator = "Strep", denominator = "Mock", response_variable = "Metabolite", Factor = "Treatment", log_transform = TRUE, p_adjust = "BH")
```

**PCA_plot**

Create a PCA plot

**Description**

Performs an ordination and outputs a PCA plot using a metabolomics count data frame and metabolomics metadata.
Usage

```
PCA_plot(count_data, metadata, variable, color, response_variable)
```

Arguments

- `count_data`: Metabolomics count data
- `metadata`: Metabolomics metadata
- `variable`: The independent variable you wish to compare and contrast
- `color`: String of what you want to color by. Usually should be the same as variable.
- `response_variable`: String of the response_variable, usually should be "Metabolite"

Examples

```
PCA_plot(count_data = c57_nos2KO_mouse_countDF, metadata = c57_nos2KO_mouse_metadata, variable = "Treatment", color = "Treatment", response_variable = "Metabolite")
```

---

pie_chart

Create a pie chart

Description

Creates a pie chart as ggplot2 object using the output from ra_table.

Usage

```
pie_chart(ratio_data, variable, column, color)
```

Arguments

- `ratio_data`: a dataframe object of percents. output from ra_table function
- `variable`: The metadata variable you are measuring, i.e. "Class"
- `column`: either "Increase", "Decrease", or "Significant_Changes"
- `color`: string denoting color for outline. use NA for no outline

Examples

```
c57_nos2KO_mouse_countDF <- assign_hierarchy(c57_nos2KO_mouse_countDF, TRUE, "KEGG")

t_test_df <- omu_summary(count_data = c57_nos2KO_mouse_countDF, metadata = c57_nos2KO_mouse_metadata, numerator = "Strep", denominator = "Mock", response_variable = "Metabolite", factor = "Treatment", log_transform = TRUE, p_adjust = "BH")

fold_change_counts <- count_fold_changes(count_data = t_test_df, "Class")
```
plate_omelette

Description

plate_omelette Internal method for KEGG_Gather

Usage

plate_omelette(count_data)

## S3 method for class 'rxn'
plate_omelette(count_data)

## S3 method for class 'genes'
plate_omelette(count_data)

Arguments

count_data The metabolomics count dataframe

plate_omelette_rxnko Clean up orthology metadata

Description

Internal function for KEGG_Gather.rxn method KEGG_Gather.rxn requires dispatch on multiple elements, so there was no way to incorporate as a method

Usage

plate_omelette_rxnko(count_data, matrix)

Arguments

count_data Metabolomics count data
matrix the matrix of KEGG metadata
**plot_bar**  
Create a bar plot

**Description**
Creates a ggplot2 object using the output file from the count_fold_changes function

**Usage**
```r
plot_bar(fc_data, fill, size, color)
```

**Arguments**
- `fc_data`: The output file from Count_Fold_Changes
- `fill`: A character vector of length 2 containing colors for filling the bars, the first color is for the "Decrease" bar while the second is for "Increase"
- `size`: A character vector of 2 numbers for the size of the bar outlines.
- `color`: A character vector of length 2 containing colors for the bar outlines

**Examples**
```r
c57_nos2KO_mouse_countDF <- assign_hierarchy(c57_nos2KO_mouse_countDF, TRUE, "KEGG")

t_test_df <- omu_summary(count_data = c57_nos2KO_mouse_countDF, 
metadata = c57_nos2KO_mouse_metadata, numerator = "Strep", denominator = "Mock", 
response_variable = "Metabolite", Factor = "Treatment", 
log_transform = TRUE, p_adjust = "BH")

fold_change_counts <- count_fold_changes(count_data = t_test_df, "Class", 
column = "Class", sig_threshold = 0.05, keep_unknowns = FALSE)

plot_bar(fc_data = fold_change_counts, fill = c("firebrick2", "dodgerblue2"),
color = c("black", "black"), size = c(1,1))
```

---

**plot_volcano**  
Create a volcano plot

**Description**
Creates a volcano plot as ggplot2 object using the output of omu_summary

**Usage**
```r
plot_volcano(count_data, column, size, strpattern, fill, sig_threshold, 
alpha, shape, color)
```
Arguments

- `count_data`: The output file from the omu_summary function.
- `column`: The column with metadata you want to highlight points in the plot with, i.e. "Class".
- `size`: Size of the points in the plot.
- `strpattern`: A character vector of levels of the column you want the plot to focus on, i.e. `strpattern = c("Carbohydrates", "Organicacids")`
- `fill`: A character vector of colors you want your points to be. Levels of a factor are organized alphabetically. All levels not in the `strpattern` argument will be set to NA.
- `sig_threshold`: An integer. Creates a horizontal dashed line for a significance threshold. i.e. `sig_threshold = 0.05`. Default value is 0.05
- `alpha`: A character vector for setting transparency of factor levels.
- `shape`: A character vector for setting the shapes for your column levels. See ggplot2 for an index of shape values.
- `color`: A character vector of colors for the column levels. If you choose to use shapes with outlines, this list will set the outline colors.

Examples

```r
c57_nos2KO_mouse_countDF <- assign_hierarchy(c57_nos2KO_mouse_countDF, TRUE, "KEGG")

t_test_df <- omu_summary(count_data = c57_nos2KO_mouse_countDF, metadata = c57_nos2KO_mouse_metadata, numerator = "Strep", denominator = "Mock", response_variable = "Metabolite", Factor = "Treatment", log_transform = TRUE, p_adjust = "BH")

plot_volcano(count_data = t_test_df, column = "Class", strpattern = c("Carbohydrates"), fill = c("firebrick", "white"), sig_threshold = 0.05, alpha = c(1,1), shape = c(1,24), color = c("black", "black"), size = 2)

plot_volcano(count_data = t_test_df, sig_threshold = 0.05, size = 2)
```

---

**ra_table**

Creates a ratio table from the count_fold_changes function output.

Description

Create a ratio table

Usage

```r
ra_table(fc_data, variable)
```
read_metabo

Import a metabolomics count data frame

Description
Wrapper for read.csv that appends the "cpd" class and sets blank cells to NA. Used to import metabolomics count data into R.

Usage
read_metabo(filepath)

Arguments
filepath a file path to your metabolomics count data

Examples
filepath_to_yourdata = paste0(system.file(package = "omu"), "/extdata/read_metabo_test.csv")
count_data <- read_metabo(filepath_to_yourdata)
Index

*Topic datasets
  c57_nos2KO_mouse_countDF, 3
  c57_nos2KO_mouse_metadata, 3

assign_hierarchy, 2

c57_nos2KO_mouse_countDF, 3
c57_nos2KO_mouse_metadata, 3
count_fold_changes, 4

KEGG_gather, 4

make_omelette, 5

omu_anova, 6
omu_summary, 7

PCA_plot, 7
pie_chart, 8
plate_omelette, 9
plate_omelette_rxnko, 9
plot_bar, 10
plot_volcano, 10

ra_table, 11
read_metabo, 12