Package: MetAlyzer (via r-universe)

September 16, 2024

Type Package

Title Read and Analyze 'MetIDQ™' Software Output Files

Version 1.0.0

Maintainer Nils Mechtel <nils.mech@gmail.com>

Depends R (>= 4.0.0)

Imports SummarizedExperiment, openxlsx, stringr, dplyr, tidyr, tibble, agricolae, ggplot2, ggrepel, utils, rlang, data.table, S4Vectors, viridis, viridisLite, plotly, limma

Description The 'MetAlyzer' S4 object provides methods to read and reformat metabolomics data for convenient data handling, statistics and downstream analysis. The resulting format corresponds to input data of the Shiny app 'MetaboExtract' (<https://www.metaboextract.shiny.dkfz.de/MetaboExtract/>).

License GPL-3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.1

Suggests rmarkdown, knitr

VignetteBuilder knitr

URL https://github.com/nilsmechtel/MetAlyzer

BugReports https://github.com/nilsmechtel/MetAlyzer/issues

Repository https://nilsmechtel.r-universe.dev

RemoteUrl https://github.com/nilsmechtel/metalyzer

RemoteRef HEAD

RemoteSha f7bbf475c366bb647548317c23bebd5fb7553611

Contents

`aggregatedData` *Get Aggregated Data*

Description

This function returns the tibble "aggregated_data".

Usage

```
aggregatedData(metalyzer_se)
```

Arguments

`metalyzer_se`	SummarizedExperiment

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_extraction_data())

aggregatedData(metalyzer_se)
```

`calculate_anova` *One-way ANOVA*

Description

This method performs a one-way ANOVA on the grouped aggregated_data (the categorical variable is removed from grouping first). The vector of the categorical variable needs to have at least two levels after removing NAs from the dependent variable vector. Otherwise a vector of NA is returned. A Tukey post-hoc test is then used to determine group names, starting with "A" followed by further letters. These group names are added to aggregated_data in the column ANOVA_Group. Thereby, metabolites can be identified which are significantly higher in one or more of the categorical variable compared to all other for each metabolite.

Usage

```
calculate_anova(
  metalyzer_se,
  categorical,
  groups = NULL,
  impute_perc_of_min = 0.2,
  impute_NA = TRUE
)
```

Arguments

`metalyzer_se`	A Metalyzer object
`categorical`	A column defining the categorical variable
`groups`	A vector of column names of aggregated_data to calculate the ANOVA group wise. If the column does not exists in aggregated_data it is automatically added from meta data. The default value is set to NULL, which uses the existing grouping of aggregated_data.
`impute_perc_of_min`	A numeric value below 1
`impute_NA`	Logical value whether to impute NA values

Value

A data frame containing the log2 fold change for each metabolite

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_extraction_data())
metalyzer_se <- renameMetaData(
  metalyzer_se,
  Extraction_Method = "Sample Description"
)
# reduced to only 'Acylcarnitines' (first metabolic class) for simplicity
drop_vec = unique(metalyzer_se@elementMetadata$metabolic_classes)[2:24]
metalyzer_se <- filterMetabolites(
  metalyzer_se,
  drop_metabolites = drop_vec
)
metalyzer_se <- filterMetaData(
  metalyzer_se,
  Tissue == "Drosophila"
)
metalyzer_se <- calculate_anova(
  metalyzer_se,
  categorical = "Extraction_Method",
  groups = c("Metabolite"),
  impute_perc_of_min = 0.2,
  impute_NA = TRUE
)
```

---

calculate_cv *Add mean, SD and CV*

Description

This function calculates the mean, standard deviation (SD) and the coefficient of variation (CV) for each group and adds them to aggregated_data.

Usage

```
calculate_cv(
  metalyzer_se,
  groups = NULL,
  cv_thresholds = c(0.1, 0.2, 0.3),
  na.rm = TRUE
)
```

Arguments

metalyzer_se	A Metalyzer object
groups	A vector of column names of aggregated_data to calculate mean, SD and CV group wise. If the column does not exists in aggregated_data it is automatically added from meta data. The default value is set to NULL, which uses the existing grouping of aggregated_data.

cv_thresholds	A numeric vector of upper thresholds (CV <= t) between 0 and 1 for CV categorization.
na.rm	a logical evaluating to TRUE or FALSE indicating whether NA values should be stripped before the computation proceeds.

Value

An updated aggregated_data tibble data frame

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_extraction_data())
metalyzer_se <- renameMetaData(
  metalyzer_se,
  Extraction_Method = "Sample Description"
)
metalyzer_se <- filterMetaData(
  metalyzer_se,
  Tissue == "Drosophila"
)
metalyzer_se <- calculate_cv(
  metalyzer_se,
  groups = c("Tissue", "Extraction_Method", "Metabolite"),
  cv_thresholds = c(0.1, 0.2, 0.3),
  na.rm = TRUE
)
```

---

calculate_log2FC *Calculate log2 fold change*

Description

This function calculates log2(FC), p-values, and adjusted p-values of the data using limma.

Usage

```
calculate_log2FC(
  metalyzer_se,
  categorical,
  impute_perc_of_min = 0.2,
  impute_NA = FALSE
)
```

Arguments

metalyzer_se	A Metalyzer object
categorical	A column specifying the two groups
impute_perc_of_min	A numeric value below 1
impute_NA	Logical value whether to impute NA values

Value

A data frame containing the log2 fold change for each metabolite

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_mutation_data_xl())
metalyzer_se <- filterMetabolites(
  metalyzer_se,
  drop_metabolites = "Metabolism Indicators"
)
metalyzer_se <- renameMetaData(
  metalyzer_se,
  Mutant_Control = "Sample Description"
)

metalyzer_se <- calculate_log2FC(
  metalyzer_se,
  categorical = "Mutant_Control",
  impute_perc_of_min = 0.2,
  impute_NA = FALSE
)
```

---

example_extraction_data

Get example extraction data

Description

This function returns the extraction_data_MxP_Quant_500.xlsx file path.

Usage

```
example_extraction_data()
```

Value

extraction_data_MxP_Quant_500.xlsx file path

Examples

```
fpath <- example_extraction_data()
```

example_meta_data *Get example meta data*

Description

This function returns the data frame loaded from example_meta_data.RDS.

Usage

```
example_meta_data()
```

Value

data frame loaded from example_meta_data.RDS

Examples

```
fpath <- example_meta_data()
```

example_mutation_data_xl *Get example mutation data*

Description

This function returns the mutation_data_MxP_Quant_500_XL.xlsx file path.

Usage

```
example_mutation_data_xl()
```

Value

mutation_data_MxP_Quant_500_XL.xlsx file path

Examples

```
fpath <- example_mutation_data_xl()
```

exportConcValues *Export filtered raw data as csv*

Description

This function exports the filtered raw data in the CSV format.

Usage

```
exportConcValues(metalyzer_se, ..., file_path = "metabolomics_data.csv")
```

Arguments

metalyzer_se	SummarizedExperiment
...	Additional columns from meta_data
file_path	file path

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_extraction_data())

output_file <- file.path(tempdir(), "metabolomics_data.csv")
exportConcValues(
  metalyzer_se,
  `Sample Description`,
  Tissue,
  file_path = output_file
)
unlink(output_file)
```

filterMetabolites *Filter metabolites*

Description

This function filters out certain classes or metabolites of the metabolites vector. If aggregated_data is not empty, metabolites and class will also be filtered here.

Usage

```
filterMetabolites(
  metalyzer_se,
  drop_metabolites = c("Metabolism Indicators"),
  drop_NA_concentration = FALSE,
  drop_quant_status = NULL,
  min_percent_valid = NULL,
```

```
  valid_status = c("Valid", "LOQ"),
  per_group = NULL,
  inplace = FALSE
)
```

Arguments

metalyzer_se	SummarizedExperiment
drop_metabolites	A character vector defining metabolite classes or individual metabolites to be removed
drop_NA_concentration	A boolean whether to drop metabolites which have any NAs in their concentration value
drop_quant_status	A character, vector of characters or list of characters specifying which quantification status to remove. Metabolites with at least one quantification status of this vector will be removed.
min_percent_valid	A numeric lower threshold between 0 and 1 (t less than or equal to x) to remove invalid metabolites that do not meet a given percentage of valid measurements per group (default per Metabolite).
valid_status	A character vector that defines which quantification status is considered valid.
per_group	A character vector of column names from meta_data that will be used to split each metabolite into groups. The threshold 'min_percent_valid' will be applied for each group. The selected columns from meta_data will be added to aggregated_data.
inplace	If FALSE, return a copy. Otherwise, do operation inplace and return None.

Value

An updated SummarizedExperiment

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_extraction_data())

drop_metabolites <- c("C0", "C2", "C3", "Metabolism Indicators",
  inplace = TRUE
)
metalyzer_se <- filterMetabolites(metalyzer_se, drop_metabolites)
# or
filterMetabolites(metalyzer_se, drop_metabolites, inplace = TRUE)
```

filterMetaData *Filter meta data*

Description

This function updates the "Filter" column in meta_data to filter out samples.

Usage

```
filterMetaData(metalyzer_se, ..., inplace = FALSE)
```

Arguments

metalyzer_se	SummarizedExperiment
...	Use ´col_name´ and condition to filter selected variables.
inplace	If FALSE, return a copy. Otherwise, do operation inplace and return None.

Value

An updated SummarizedExperiment

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_extraction_data())

metalyzer_se <- filterMetaData(metalyzer_se, !is.na(Tissue))
metalyzer_se <- filterMetaData(metalyzer_se, `Sample Description` %in% 1:6)
# or
filterMetaData(metalyzer_se, !is.na(Tissue), inplace = TRUE)
filterMetaData(metalyzer_se, `Sample Description` %in% 1:6, inplace = TRUE)
```

log2FC *Get log2FC Data*

Description

This function returns the tibble "log2FC".

Usage

```
log2FC(metalyzer_se)
```

Arguments

metalyzer_se	SummarizedExperiment

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_mutation_data_xl())
metalyzer_se <- filterMetabolites(
  metalyzer_se,
  drop_metabolites = "Metabolism Indicators"
)
metalyzer_se <- renameMetaData(
  metalyzer_se,
  Mutant_Control = "Sample Description"
)

metalyzer_se <- calculate_log2FC(
  metalyzer_se,
  categorical = "Mutant_Control",
  impute_perc_of_min = 0.2,
  impute_NA = TRUE
)

log2FC(metalyzer_se)
```

`metalyzer_colors` *Get MetAlyzer colors*

Description

This function returns the vector loaded from metalyzer_colors.RDS.

Usage

```
metalyzer_colors()
```

Value

data frame loaded from metalyzer_colors.RDS

Examples

```
fpath <- metalyzer_colors()
```

MetAlyzer_dataset *Open file and read data*

Description

This function creates a SummarizedExperiment (SE) from the given 'MetIDQ' output Excel sheet: metabolites (rowData), meta data (colData), concentration data (assay), quantification status(assay) The column "Sample Type" and the row "Class" are used as anchor cells in the Excel sheet and are therefore a requirement.

Usage

```
MetAlyzer_dataset(
  file_path,
  sheet = 1,
 status_list = list(Valid = c("#B9DE83", "#00CD66"), LOQ = c("#B2D1DC", "#7FB2C5",
   "#87CEEB"), LOD = c("#A28BA3", "#6A5ACD"), `ISTD Out of Range` = c("#FFF099",
    "#FFFF33"), Invalid = "#FFFFCC", Incomplete = c("#CBD2D7", "#FFCCCC")),
  silent = FALSE
)
```

Arguments

file_path	A character specifying the file path to the Excel file.
sheet	A numeric index specifying which sheet of the Excel file to use.
status_list	A list of HEX color codes for each quantification status.
silent	If TRUE, mute any print command.

Value

A Summarized Experiment object

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_extraction_data())
```

pathway *Get pathway file path*

Description

This function returns the pathway.xlsx file path.

Usage

```
pathway()
```

Value

pathway.xlsx file path

Examples

```
fpath <- pathway()
```

---

plotly_network *Plotly Log2FC Network Plot*

Description

This function returns a list with interactive networkplot based on log2 fold change data.

Usage

```
plotly_network(
  metalyzer_se,
  q_value = 0.05,
  metabolite_node_size = 11,
  connection_width = 1.25,
  pathway_text_size = 20,
  pathway_width = 10,
  plot_height = 800
)
```

Arguments

metalyzer_se	A MetAlyzer Object
q_value	A numeric value specifying the cutoff for q-value
metabolite_node_size	The text size of the metabolite Nodes
connection_width	The line width of connections between metabolites
pathway_text_size	The text size of pathway annotations
pathway_width	The line width of pathway-specific connection coloring
plot_height	The height of the Plot in pixel [px]

Value

plotly object

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_mutation_data_xl())
metalyzer_se <- filterMetabolites(
  metalyzer_se,
  drop_metabolites = "Metabolism Indicators"
)
metalyzer_se <- renameMetaData(
  metalyzer_se,
  Mutant_Control = "Sample Description"
)

metalyzer_se <- calculate_log2FC(
  metalyzer_se,
  categorical = "Mutant_Control",
  impute_perc_of_min = 0.2,
  impute_NA = FALSE
)

p_network <- plotly_network(metalyzer_se, q_value = 0.05)
```

---

plotly_scatter *Plotly Log2FC Scatter Plot*

Description

This function returns a list with an interactive scatterplot based on log2 fold change data and a comprehensive Legend.

Usage

```
plotly_scatter(
  metalyzer_se,
 signif_colors = c(`#5F5F5F` = 1, `#FEBF6E` = 0.1, `#EE5C42` = 0.05, `#8B1A1A` = 0.01),
  class_colors = metalyzer_colors()
)
```

Arguments

metalyzer_se	A Metalyzer object
signif_colors	signif_colors
class_colors	A csv file containing class colors hexcodes

Value

plotly object

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_mutation_data_xl())
metalyzer_se <- filterMetabolites(
  metalyzer_se,
  drop_metabolites = "Metabolism Indicators"
)
metalyzer_se <- renameMetaData(
  metalyzer_se,
  Mutant_Control = "Sample Description"
)
metalyzer_se <- calculate_log2FC(
  metalyzer_se,
  categorical = "Mutant_Control",
  impute_perc_of_min = 0.2,
  impute_NA = TRUE
)

p_scatter <- plotly_scatter(metalyzer_se)
```

---

plotly_vulcano *Plotly Log2FC Vulcano Plot*

Description

This function returns a list with interactive vulcanoplot based on log2 fold change data.

Usage

```
plotly_vulcano(
  metalyzer_se,
  cutoff_y = 0.05,
  cutoff_x = 1.5,
  class_colors = metalyzer_colors()
)
```

Arguments

metalyzer_se	A Metalyzer object
cutoff_y	A numeric value specifying the cutoff for q-value
cutoff_x	A numeric value specifying the cutoff for log2 fold change
class_colors	A csv file containing class colors hexcodes

Value

plotly object

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_mutation_data_xl())
metalyzer_se <- filterMetabolites(
  metalyzer_se,
  drop_metabolites = "Metabolism Indicators"
)
metalyzer_se <- renameMetaData(
  metalyzer_se,
  Mutant_Control = "Sample Description"
)
metalyzer_se <- calculate_log2FC(
  metalyzer_se,
  categorical = "Mutant_Control",
  impute_perc_of_min = 0.2,
  impute_NA = TRUE
)

p_vulcano <- plotly_vulcano(metalyzer_se,
                        cutoff_y = 0.05,
                        cutoff_x = 1.5)
```

---

plot_log2FC *Plot log2 fold change*

Description

This method plots the log2 fold change for each metabolite.

Usage

```
plot_log2FC(
  metalyzer_se,
  signif_colors = c(`#5F5F5F` = 1, `#FEBF6E` = 0.1, `#EE5C42` = 0.05, `#8B1A1A` = 0.01),
  hide_labels_for = c(),
  class_colors = "MetAlyzer",
  polarity_file = "MxPQuant500",
  vulcano = FALSE
)
```

Arguments

metalyzer_se	A Metalyzer object
signif_colors	signif_colors
hide_labels_for	vector of Metabolites or Classes for which no labels are printed
class_colors	class_colors
polarity_file	polarity_file
vulcano	boolean value to plot a vulcano plot

Value

ggplot object

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_mutation_data_xl())
metalyzer_se <- filterMetabolites(
  metalyzer_se,
  drop_metabolites = "Metabolism Indicators"
)
metalyzer_se <- renameMetaData(
  metalyzer_se,
  Mutant_Control = "Sample Description"
)
metalyzer_se <- calculate_log2FC(
  metalyzer_se,
  categorical = "Mutant_Control",
  impute_perc_of_min = 0.2,
  impute_NA = TRUE
)

# p_vulcano <- plot_log2FC(metalyzer_se, vulcano=TRUE)
# p_fc <- plot_log2FC(metalyzer_se, vulcano=FALSE)
```

---

plot_network *Plot Pathway Network*

Description

This function plots the log2 fold change for each metabolite and visualizes it, in a pathway network.

Usage

```
plot_network(
  metalyzer_se,
  q_value = 0.05,
  metabolite_text_size = 3,
  connection_width = 0.75,
  pathway_text_size = 6,
  pathway_width = 4,
  scale_colors = c("green", "black", "magenta")
)
```

Arguments

metalyzer_se	A Metalyzer object
q_value	The q-value threshold for significance

`metabolite_text_size`
: The text size of metabolite labels

`connection_width`
: The line width of connections between metabolites

`pathway_text_size`
: The text size of pathway annotations

`pathway_width`
: The line width of pathway-specific connection coloring

`scale_colors`
: A vector of length 3 with colors for low, mid and high of the gradient.

Value

ggplot object

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_mutation_data_xl())
metalyzer_se <- filterMetabolites(
  metalyzer_se,
  drop_metabolites = "Metabolism Indicators"
)
metalyzer_se <- renameMetaData(
  metalyzer_se,
  Mutant_Control = "Sample Description"
)

metalyzer_se <- calculate_log2FC(
  metalyzer_se,
  categorical = "Mutant_Control",
  impute_perc_of_min = 0.2,
  impute_NA = FALSE
)

network <- plot_network(metalyzer_se, q_value = 0.05)
```

---

polarity *Get polarity file path*

Description

This function returns the polarity.csv file path.

Usage

```
polarity()
```

Value

polarity.csv file path

Examples

```
fpath <- polarity()
```

`read_named_region` *Read Named Regions*

Description

This function reads in the named regions of an excel file.

Usage

```
read_named_region(file_path, named_region)
```

Arguments

`file_path`	The file path of the file
`named_region`	The region name u want to read in

`renameMetaData` *Rename meta data*

Description

This function renames a column of meta_data.

Usage

```
renameMetaData(metalyzer_se, ..., inplace = FALSE)
```

Arguments

`metalyzer_se`	SummarizedExperiment
`...`	Use new_name = old_name to rename selected variables
`inplace`	If FALSE, return a copy. Otherwise, do operation inplace and return None.

Value

An updated SummarizedExperiment

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_extraction_data())

metalyzer_se <- renameMetaData(
  metalyzer_se,
  Method = `Sample Description`
)
# or
renameMetaData(metalyzer_se, Model_Organism = Tissue, inplace = TRUE)
```

---

summarizeConcValues *Summarize concentration values*

Description

This function prints quantiles and NAs of raw data.

Usage

```
summarizeConcValues(metalyzer_se)
```

Arguments

metalyzer_se	SummarizedExperiment

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_extraction_data())

summarizeConcValues(metalyzer_se)
```

---

summarizeQuantData *Summarize quantification status*

Description

This function lists the number of each quantification status and its percentage.

Usage

```
summarizeQuantData(metalyzer_se)
```

Arguments

metalyzer_se	SummarizedExperiment

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_extraction_data())

summarizeQuantData(metalyzer_se)
```

`updateMetaData` *Update meta data*

Description

This function adds another column to filtered meta_data.

Usage

```
updateMetaData(metalyzer_se, ..., inplace = FALSE)
```

Arguments

`metalyzer_se`	SummarizedExperiment
`...`	Use ´new_col_name = new_column´ to rename selected variables
`inplace`	If FALSE, return a copy. Otherwise, do operation inplace and return None.

Value

An updated SummarizedExperiment

Examples

```
metalyzer_se <- MetAlyzer_dataset(file_path = example_extraction_data())

metalyzer_se <- updateMetaData(
  metalyzer_se,
  Date = Sys.Date(), Analyzed = TRUE
)
# or
updateMetaData(
  metalyzer_se,
  Date = Sys.Date(), Analyzed = TRUE, inplace = TRUE
)
```

Index