Package: MetAlyzer (via r-universe)

May 27, 2026

Type Package

Title Read and Analyze 'MetIDQ™' Software Output Files

Version 1.2.0

Maintainer Junyan Lu <junyan.lu@uni-heidelberg.de>

Depends R (>= 4.0.0)

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

Suggests bslib, devtools, DT, ggpubr, htmlwidgets, kableExtra, knitr, remotes, rmarkdown, shinyBS, shinycssloaders, shinyWidgets, svglite, writexl

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 7.3.3

VignetteBuilder knitr

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

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

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev zlib1g-dev

Repository https://lu-group-ukhd.r-universe.dev

Date/Publication 2026-04-27 08:43:43 UTC

RemoteUrl https://github.com/lu-group-ukhd/metalyzer

RemoteRef HEAD

RemoteSha 793743572dd95ee6e460747c42a6e240e91efb4d

Contents

`aggregated_data` *Get Aggregated Data*

Description

This function returns the tibble "aggregated_data".

Usage

```
aggregated_data(metalyzer_se)
```

Arguments

`metalyzer_se`	SummarizedExperiment

Examples

```
metalyzer_se <- MetAlyzer::read_webidq(file_path = MetAlyzer::load_demodata_biocrates())

MetAlyzer::aggregated_data(metalyzer_se)
```

`calculate_conditional_mean` *Helper function to calculate conditional mean based on significance*

Description

Calculates mean of a metric vector based on q-values. Prioritizes values where qval <= q_thresh. If none exist, uses all non-NA values.

Usage

```
calculate_conditional_mean(metric_vec, qval_vec, q_thresh)
```

Arguments

`metric_vec`	Numeric vector of the metric to average (e.g., log2FC).
`qval_vec`	Numeric vector of corresponding q-values.
`q_thresh`	Significance threshold for q-values.

Value

The calculated conditional mean, or NA.

`calculate_node_aggregates_conditional` *Calculate Node-Level Aggregate Statistics (Conditional on Significance)*

Description

Calculates mean log2FC, p-value, and q-value for each node (Label), prioritizing significant metabolites (qval <= q_value). If none are significant, uses all measured metabolites for the node. Adds results to both dataframes.

Usage

```
calculate_node_aggregates_conditional(
  nodes_sep_df,
  nodes_orig_df,
  q_value,
  stat_col_name,
  ...
)
```

Arguments

nodes_sep_df	Dataframe with metabolites separated (e.g., 'nodes_final'). Must contain Label, log2FC, qval.
nodes_orig_df	Original dataframe with potentially semi-colon separated metabolites. Must contain Label.
q_value	Significance threshold for q-values (e.g., 0.05).
stat_col_name	p value column name
...	Column names of numeric values to be processed (e.g., log2FC, pval, qval).

Value

A list containing two dataframes: $nodes_separated: Input nodes_sep_df with 2 new columns: node_values, node_stat $nodes: Input nodes_orig_df with 2 new columns: node_values, node_stat

---

create_viridis_style *Creates a viridis color style for Plotly plots.*

Description

This function can generate either a Plotly-compatible colorscale for a color bar or a vector of hex color codes for manual coloring.

Usage

```
create_viridis_style(
  color_scale,
  type = "scale",
  data = NULL,
  values_col_name = NULL
)
```

Arguments

color_scale	The name of the palette (e.g., "Magma", "Viridis").
type	The desired output type: "scale" (for a color bar), "hex" or "inital" (for a color scalde, a vector of hex codes, or the correct scale for viridis package). Defaults to "scale".
data	The data frame containing the values. Only required if type = "hex".
values_col_name	The name of the column with numeric values. Only required if type = "hex".

Value

A data frame if type is "scale", or a character vector if type is "hex".

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 <- MetAlyzer::example_mutation_data_xl()
```

export_conc_values

Export filtered raw data as csv

Description

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

Usage

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

Arguments

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

Examples

```
metalyzer_se <- MetAlyzer::read_webidq(file_path = MetAlyzer::load_demodata_biocrates())

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

filter_meta_data *Filter meta data*

Description

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

Usage

```
filter_meta_data(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::read_webidq(file_path = MetAlyzer::load_demodata_biocrates())

metalyzer_se <- MetAlyzer::filter_meta_data(metalyzer_se, `Sample Description` %in% 1:6)
```

filter_metabolites *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

```
filter_metabolites(
  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::read_webidq(file_path = MetAlyzer::load_demodata_biocrates())

drop_metabolites <- c("C0", "C2", "C3", "Metabolism Indicators",
  inplace = TRUE
)
metalyzer_se <- MetAlyzer::filter_metabolites(metalyzer_se, drop_metabolites)
```

---

load_demodata_biocrates

Get demodata from biocrates

Description

This function returns the Metalyzer_demo dataset_biocrates MxP Quant 500 XL_2025-04.xlsx file path.

Usage

```
load_demodata_biocrates()
```

Value

Metalyzer_demo dataset_biocrates MxP Quant 500 XL_2025-04 file path

Examples

```
fpath <- MetAlyzer::load_demodata_biocrates()
```

---

load_rawdata_extraction

Get example extraction data

Description

This function returns the extraction_data_MxP_Quant_500.xlsx file path.

Usage

```
load_rawdata_extraction()
```

Value

extraction_data_MxP_Quant_500.xlsx file path

Examples

```
fpath <- MetAlyzer::load_rawdata_extraction()
```

---

log2FC

Get log2FC Data

Description

This function returns the tibble "log2FC".

Usage

```
log2FC(metalyzer_se)
```

Arguments

metalyzer_se	SummarizedExperiment

Examples

```
metalyzer_se <- MetAlyzer::read_webidq(file_path = MetAlyzer::load_demodata_biocrates())
metalyzer_se@metadata$log2FC <- readRDS(MetAlyzer::toy_diffres())
MetAlyzer::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::metalyzer_colors()
```

MetAlyzer_dataset *Open file and read data*

Description

This function was deprecated in version v2.0.0

Usage

```
MetAlyzer_dataset(...)
```

Arguments

...	Declare this function as out of date

pathway *Get pathway file path*

Description

This function returns the latest pathway.xlsx file path.

Usage

```
pathway()
```

Value

pathway.xlsx file path

Examples

```
fpath <- MetAlyzer::pathway()
```

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(
  log2fc_df,
  q_value = 0.05,
  metabolite_col_name = "Metabolite",
  values_col_name = "log2FC",
  stat_col_name = "qval",
  metabolite_text_size = 3,
  connection_width = 0.75,
  pathway_text_size = 6,
  pathway_width = 3,
  exclude_pathways = NULL,
  color_scale = "Viridis",
  gradient_colors = NULL,
  save_as = NULL,
  folder_name = format(Sys.Date(), "%Y-%m-%d"),
  folder_path = NULL,
  file_name = "network",
  format = "pdf",
```

```
  width = 29.7,
  height = 21,
  units = "cm",
  overwrite = FALSE
)
```

Arguments

log2fc_df	A dataframe with log2FC, qval, additional columns
q_value	The q-value threshold for significance
metabolite_col_name	Columnname that holds the Metabolites
values_col_name	Column name of a column that holds numeric values, to be plotted **Default = "log2FC"**
stat_col_name	Columnname that holds numeric stat values that are used for significance **Default = "qval"**
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
exclude_pathways	Pathway names that are exluded from plotting
color_scale	A string specifying the color scale to use. Options include ‘"Viridis"‘, ‘"Plasma"‘, ‘"Magma"‘, ‘"Inferno"‘, ‘"Cividis"‘, ‘"Rocket"‘, ‘"Mako"‘, and ‘"Turbo"‘, which use the ‘viridis‘ color scales. If ‘"gradient"‘ is selected, a custom gradient is applied based on ‘gradient_colors‘.
gradient_colors	A vector of length 2 or 3 specifying the colors for a custom gradient. If two colors are provided (‘c(low, high)‘), ‘scale_fill_gradient()‘ is used. If three colors are provided (‘c(low, mid, high)‘), ‘scale_fill_gradient2()‘ is used. If ‘NULL‘ or incorrectly specified, the viridis color scale is applied.
save_as	*Optional:* Select the file type of output plots. Options are svg, pdf, png or NULL. **Default = "NULL"**
folder_name	Name of the folder where the plot will be saved. Special characters will be removed automatically. **Default = date**
folder_path	*Optional:* User-defined path where the folder should be created. If not provided, results will be saved in ‘MetAlyzer_results‘ within the working directory. **Default = NULL**
file_name	Name of the output file (without extension). **Default = "network"**
format	File format for saving the plot (e.g., "png", "pdf", "svg"). **Default = "pdf"**
width	Width of the saved plot in specified units. **Default = 29.7**

height	Height of the saved plot in specified units. **Default = 21.0**
units	Units for width and height (e.g., "in", "cm", "mm"). **Default = "cm"**
overwrite	Logical: If 'TRUE', overwrite existing files without asking. If 'FALSE', prompt user before overwriting. **Default = FALSE**

Value

list with ggplot object and table of node summaries

Examples

```
log2fc_df <- readRDS(MetAlyzer::toy_diffres())
network <- MetAlyzer::plot_network(log2fc_df, q_value = 0.05)
network$Plot
network$Table
```

---

plot_scatter *Scatter Plot Visualization*

Description

This method creates a scatter plot of the log2 fold change for each metabolite.

Usage

```
plot_scatter(
  log2fc_df,
  show_labels_for = NULL,
  values_col_name = "log2FC",
  stat_col_name = "qval",
  show_p_value = TRUE,
 signif_colors = c(`#5F5F5F` = 1, `#FEBF6E` = 0.1, `#EE5C42` = 0.05, `#8B1A1A` = 0.01),
  save_as = NULL,
  folder_name = format(Sys.Date(), "%Y-%m-%d"),
  folder_path = NULL,
  file_name = "network",
  format = "pdf",
  width = 29.7,
  height = 21,
  units = "cm",
  overwrite = FALSE
)
```

Arguments

log2fc_df	DF with metabolites as row names and columns including log2FC, Class, qval columns.
show_labels_for	Vector with Strings of Metabolite names or classes.
values_col_name	Column name of a column that holds numeric values, to be plotted **Default = "log2FC"**
stat_col_name	Columnname that holds numeric stat values that are used for significance **Default = "qval"**
show_p_value	Boolean Value, to color p-values according to their significance level and add a Legend **Default = TRUE**
signif_colors	Vector assigning significance values different colors
save_as	*Optional:* Select the file type of output plots. Options are svg, pdf, png or NULL. **Default = "NULL"**
folder_name	Name of the folder where the plot will be saved. Special characters will be removed automatically. **Default = date**
folder_path	*Optional:* User-defined path where the folder should be created. If not provided, results will be saved in 'MetAlyzer_results' within the working directory. **Default = NULL**
file_name	Name of the output file (without extension). **Default = "network"**
format	File format for saving the plot (e.g., "png", "pdf", "svg"). **Default = "pdf"**
width	Width of the saved plot in specified units. **Default = 29.7**
height	Height of the saved plot in specified units. **Default = 21.0**
units	Units for width and height (e.g., "in", "cm", "mm"). **Default = "cm"**
overwrite	Logical: If 'TRUE', overwrite existing files without asking. If 'FALSE', prompt user before overwriting. **Default = FALSE**

Value

ggplot object

Examples

```
log2fc_df <- readRDS(MetAlyzer::toy_diffres())
scatter <- MetAlyzer::plot_scatter(log2fc_df)
```

polarity *Get polarity file path*

Description

This function returns the polarity.csv file path.

Usage

```
polarity()
```

Value

polarity.csv file path

Examples

```
fpath <- MetAlyzer::polarity()
```

read_edges *Read and Validate Network Edges (Connections)*

Description

Reads edge data from a specified named region, validates that connected nodes exist and are not self-loops, and removes invalid edges.

Usage

```
read_edges(network_file, nodes, pathways, region_name = "Connections_Header")
```

Arguments

network_file	Path to the input file containing edge data.
nodes	A data frame of validated nodes (output of read_nodes).
pathways	A data frame of validated pathways (output of read_pathways).
region_name	The named region or sheet containing connections header info.

Value

A data frame of validated edges.

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

read_nodes *Read and Validate Network Nodes (Metabolites)*

Description

Reads node data from a specified named region, validates entries against pathway information, cleans labels, removes invalid nodes, and sets row names.

Usage

```
read_nodes(network_file, pathways, region_name = "Metabolites_Header")
```

Arguments

network_file	Path to the input file containing node data.
pathways	A data frame of validated pathways (output of read_pathways).
region_name	The named region or sheet containing metabolite header info.

Value

A data frame of validated nodes with labels as row names.

read_pathways *Read and Validate Pathway Annotations*

Description

Reads pathway data from a specified named region in the pathway file, validates entries, removes invalid ones, and sets row names.

Usage

```
read_pathways(network_file, region_name = "Pathways_Header")
```

Arguments

network_file	Path to the input file containing pathway data.
region_name	The named region or sheet containing pathway header info.

Value

A data frame of validated pathway annotations with labels as row names.

read_webidq *Open file and read data*

Description

This function creates a SummarizedExperiment (SE) from the given 'webidq' 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

```
read_webidq(
  file_path,
  sheet = 1,
 status_list = list(Valid = c("#B9DE83", "#00CD66"), LOQ = c("#B2D1DC", "#7FB2C5",
    "#87CEEB"), LOD = c("#A28BA3", "#6A5ACD", "#BBA7B9"), `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

```
Path <- MetAlyzer::load_demodata_biocrates()
metalyzer_se <- MetAlyzer::read_webidq(file_path = Path)
```

---

rename_meta_data *Rename meta data*

Description

This function renames a column of meta_data.

Usage

```
rename_meta_data(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::read_webidq(file_path = MetAlyzer::load_demodata_biocrates())

metalyzer_se <- MetAlyzer::rename_meta_data(
  metalyzer_se,
  Method = `Sample Description`
)
```

save_plot *Save plots*

Description

This function saves a given ggplot object to a specified folder and file format. It ensures that the folder structure exists and cleans the folder name to remove special characters.

Usage

```
save_plot(
  plot,
  folder_name = format(Sys.Date(), "%Y-%m-%d"),
  folder_path = NULL,
  file_name = "network",
  format = "pdf",
  units = "cm",
  height = 21,
  width = 29.7,
  overwrite = FALSE
)
```

Arguments

plot	A ggplot object to be saved.
folder_name	Name of the folder where the plot will be saved. Special characters will be removed automatically. **Default = date**
folder_path	*Optional:* User-defined path where the folder should be created. If not provided, results will be saved in 'MetAlyzer_results' within the working directory. **Default = NULL**
file_name	Name of the output file (without extension). **Default = "network"**
format	File format for saving the plot (e.g., "png", "pdf", "svg"). **Default = "pdf"**
units	Units for width and height (e.g., "in", "cm", "mm"). **Default = "cm"**
height	Height of the saved plot in specified units. **Default = 21.0**
width	Width of the saved plot in specified units. **Default = 29.7**
overwrite	Logical: If 'TRUE', overwrite existing files without asking. If 'FALSE', prompt user before overwriting. **Default = FALSE**

Value

The function does not return anything but saves the plot to the specified directory.

start_app *Launch the Shiny App*

Description

This function launches the Shiny application.

Usage

```
start_app()
```

summarize_conc_values *Summarize concentration values*

Description

This function prints quantiles and NAs of raw data.

Usage

```
summarize_conc_values(metalyzer_se)
```

Arguments

metalyzer_se	SummarizedExperiment

Examples

```
metalyzer_se <- MetAlyzer::read_webidq(file_path = MetAlyzer::load_demodata_biocrates())

MetAlyzer::summarize_conc_values(metalyzer_se)
```

summarize_quant_data *Summarize quantification status*

Description

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

Usage

```
summarize_quant_data(metalyzer_se)
```

Arguments

metalyzer_se	SummarizedExperiment

Examples

```
metalyzer_se <- MetAlyzer::read_webidq(file_path = MetAlyzer::load_demodata_biocrates())

MetAlyzer::summarize_quant_data(metalyzer_se)
```

---

toy_diffres *Get example log2fc data*

Description

This function returns the log2fc dataframe of the Metalyzer_demo dataset_biocrates MxP Quant 500 XL_2025-04 file, created with the MetaVizPro package.

Usage

```
toy_diffres()
```

Value

toy_diffres.rds file path

Examples

```
fpath <- MetAlyzer::toy_diffres()
```

---

update_meta_data *Update meta data*

Description

This function adds another column to filtered meta_data.

Usage

```
update_meta_data(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::read_webidq(file_path = MetAlyzer::load_demodata_biocrates())

metalyzer_se <- MetAlyzer::update_meta_data(
  metalyzer_se,
  Date = Sys.Date(), Analyzed = TRUE
)
```

Index