Package 'TockyPrep'

November 6, 2024

Version 0.1.0

Date 2024-11-05

Title Data Preprocessing for Fluorescent Timer Reporters Using the Timer-Of-Cell-Kinetics-of-activitY (Tocky)

Author Masahiro Ono [aut, cre]

Maintainer Masahiro Ono <monotockylab@gmail.com>

Description This package provides data preprocessing methods for analyzing Fluorescent Timer data obtained by flow cytometry. Specifically, it provides the trigonometric transformation of Timer fluorescence to generate Timer Angle and Timer Intensity.

Depends R (>= 4.2.0), utils, stats, graphics, grDevices

Suggests knitr, rmarkdown, KernSmooth

VignetteBuilder knitr

License Apache License 2.0

URL https://github.com/MonoTockyLab/TockyPrep, https://MonoTockyLab.github.io/TockyPrep

BugReports https://github.com/MonoTockyLab/TockyPrep/issues

Encoding UTF-8

RoxygenNote 7.3.2

Contents

`log_data` *Log-transform selected variables in a data frame*

Description

This function applies a custom log10 transformation to selected variables in a data frame. It allows optional adjustment of the data before logging and removal of rows with zeros in the transformed variables.

Usage

```
log_data(x, columns = NULL, add_offset = TRUE)
```

Arguments

`x`	A data frame containing the data to be log-transformed.
`columns`	Optional character vector specifying the names of the columns to be log-transformed. If `NULL`, the function will prompt the user to select columns interactively. Default is `NULL`.
`add_offset`	Logical value indicating whether to adjust the data before logging by subtracting the 1% quantile and adding 1. This helps in handling negative values and zeros. Default is `TRUE`.

Value

A data frame with the selected variables log-transformed and combined with the remaining variables.

Examples

```
## Not run:
# Create an example data frame
df <- data.frame(
  Var1 = rnorm(100, mean = 100, sd = 20),
  Var2 = rnorm(100, mean = 50, sd = 10),
  Var3 = rnorm(100, mean = 10, sd = 2)
)

# Log-transform Var1 and Var2 with adjustment
df_logged <- log_data(df, columns = c("Var1", "Var2"), add_offset = TRUE)

# Log-transform variables with interactive selection
df_logged <- log_data(df)

## End(Not run)
```

plot_tocky *Generate basic QC plots for Tocky data*

Description

This function visualizes either Timer fluorescence (Blue vs Red) or Timer dynamics by the Tocky method (Angle vs Intensity) based on the specified mode.

Usage

```
plot_tocky(
  timer_transform_output,
  file = "PlotTocky",
  pseudocolour = TRUE,
  pdf = FALSE,
  output = "QC",
  n = 4,
  plot_mode = "Timer fluorescence",
  lower_quantile_cutoff = 0.01,
  select = FALSE,
  group = TRUE,
  group_order = NULL,
  interactive = TRUE,
  save = FALSE,
  samplefile = NULL,
  verbose = TRUE
)
```

Arguments

timer_transform_output	A list object returned by 'timer_transform', containing 'sample_definition'.
file	File name.
pseudocolour	A logical argument for whether to use pseudocolour in plots.
pdf	A logical argument; if FALSE, a jpeg file is generated instead.
output	The output directory name for output files.
n	A number; n x n plots will be generated in the output Tocky plot file, max is 4 for 16 plots.
plot_mode	Either "Timer fluorescence" for Blue vs Red plots, "Normalized Timer fluorescence" for normalized plots, or "Timer Angle and Intensity" for Angle vs Intensity plots.
lower_quantile_cutoff	Lower quantile cutoff for setting the plot ranges in fluorescence mode.
select	Logical indicating whether to manually select samples for plotting.
group	Logical indicating whether to group plots based on the 'group' field in 'sample_definition'.
group_order	Optional character vector for specifying the order of the panels when using the group option.

interactive	Logical indicating whether to prompt the user to select plot_mode. Defaults to 'TRUE'.
save	A logical argument; if FALSE, plots are shown in an X window.
samplefile	Character vector specifying the sample files. Defaults to 'NULL'.
verbose	Logical indicating whether to print progress messages. Default is 'TRUE'.

Examples

```
## Not run:
  plot_tocky(x, plot_mode = "Timer fluorescence")
  plot_tocky(x, plot_mode = "Timer Angle and Intensity")

## End(Not run)
```

---

prep_tocky *Prepare Data for Timer Transformation Using Flow Cytometric Data*

Description

This function prepares the dataset for timer transformation analysis by identifying common variables across sample files, configuring necessary control files, and setting up variables for downstream analysis. The function supports both interactive and non-interactive file selection modes.

Usage

```
prep_tocky(path = ".", interactive = TRUE, negfile = NULL, samplefile = NULL)
```

Arguments

path	Character string specifying the directory where the data files are located. Defaults to the current directory '.'.
interactive	Logical indicating whether to prompt the user to select sample files. Defaults to 'TRUE'.
negfile	Character string specifying the negative control file. If 'NULL', the user will be prompted to select a file. Defaults to 'NULL'.
samplefile	Character vector specifying the sample files. If 'NULL' and 'samplefilechoice' is 'TRUE', the user will be prompted to select files. Defaults to 'NULL'.

Value

A list containing paths to the control file, selected sample files, and the standardized variables used in the analysis.

Examples

```
## Not run:
  # Interactive file selection
  prep_data <- prep_tocky(path='data', output='output')

  # Specifying files directly for non-interactive usage
  prep_data <- prep_tocky(
    path='data',
    output='output',
    negfile='neg_control.csv',
    samplefile=c('sample1.csv', 'sample2.csv')
  )

## End(Not run)
```

`sample_definition` *Update sample definitions and group assignments*

Description

This function takes the output from ‘timer_transform‘, specifically the ‘sample_definition‘ data frame, exports it to a CSV file for the user to edit group assignments, and then reads the updated file back into R.

Usage

```
sample_definition(
  timer_transform_output,
  sample_definition = NULL,
  output_dir = NULL,
  filename = "sample_definition.csv",
  sep = ",",
  verbose = TRUE,
  interactive = FALSE
)
```

Arguments

`timer_transform_output`	A list object returned by ‘timer_transform‘, containing ‘sample_definition‘.
`sample_definition`	(Optional) to use a data frame object as an input, specify the input object by this parameter. Defaul is ‘NULL‘.
`output_dir`	Character string specifying the directory to save the ‘sample_definition.csv‘ file. If ‘NULL‘, the file is saved in the current working directory. Default is ‘NULL‘.
`filename`	Character string specifying the name of the sample definition file. Default is ‘"sample_definition.csv"‘.
`sep`	Character string indicating the field separator used in the CSV file. Default is ‘","‘.
`verbose`	Logical indicating whether to display messages. Default is ‘TRUE‘.
`interactive`	Logical indicating whether to use an interactive session to export a file for sample grouping and enable user to edit it and import. Defaults to ‘TRUE‘.

Value

A data frame containing the updated sample definitions with user-assigned groupings.

Examples

```
## Not run:
  # Assuming `result` is the output from `timer_transform`
  sample_def <- sample_definition(result, output_dir = "output_directory")
  # The function will pause, allowing you to edit the 'group' column in the CSV file.
  # After editing and saving the file, press Enter in R to continue.
  # The updated sample definitions will be returned as a data frame.

## End(Not run)
```

`timer_transform` *Perform Timer Transformation on Flow Cytometry Data*

Description

This function processes flow cytometry data by applying FSC correction, normalization, and trigonometric transformation to the Blue and Red fluorescence channels.

Usage

```
timer_transform(
  prep,
  select = TRUE,
  blue_channel = NULL,
  red_channel = NULL,
  normalization_method = "MAD",
  red_threshold = NULL,
  blue_threshold = NULL,
  interactive_gating = FALSE,
  verbose = TRUE,
  q = 0.998,
  normalization = TRUE
)
```

Arguments

`prep`	A list containing file paths and variables, typically the output from `prep_tocky`.
`select`	Logical indicating whether to choose Timer fluorescence channels interactively. Default is 'TRUE'.
`blue_channel`	Character string specifying the Blue fluorescence channel name. If 'NULL', the function attempts to determine it automatically.
`red_channel`	Character string specifying the Red fluorescence channel name. If 'NULL', the function attempts to determine it automatically.
`normalization_method`	Charcter string specifying the normalization method to be applied to Timer fluorescence Default is 'MAD'. The alternative is 'SD'.

red_threshold	Numeric specifying the Red channel gating threshold. If 'NULL', gating is performed automatically or interactively based on 'interactive_gating'.
blue_threshold	Numeric specifying the Blue channel gating threshold. If 'NULL', gating is performed automatically or interactively based on 'interactive_gating'.
interactive_gating	Logical indicating whether to perform interactive gating when thresholds are not provided. Default is 'FALSE'.
verbose	Logical indicating whether to print progress messages. Default is 'TRUE'.
q	Quantile value used for automatic Timer thresholds. Default is 0.998.
normalization	Logical indicating whether to apply Timer fluorescence normalization. Default is 'TRUE'.

Value

A list containing:

- transformed_data: Data frame with angle, intensity, and other variables.
- normalization_parameters: List with normalization values and coefficients.
- cell_counts: Data frame with cell counts for each sample.
- sample_definition: Data frame with sample file names and placeholder group column.

Examples

```
## Not run:
  # Assuming `prep_data` is the output from `prep_tocky`
  result <- timer_transform(prep_data)

## End(Not run)
```

Index