Package 'TockyPrep'

January 16, 2025

Version 0.1.2

Date 2025-01-16

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, methods

Imports shiny

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

explore_timer_transform *Launch a Shiny App for Exploring timer_transform Parameter Space*

Description

The Shiny application provides a user interface for adjusting the blue and red fluorescence thresholds and choosing between normalization methods. It updates visualizations based on user inputs to aid in determining optimal parameters for data analysis.

Usage

```
explore_timer_transform(prep, transformed_data)
```

Arguments

prep	A prep object containing file paths and variables, typically the output from prep_tocky.
transformed_data	A TockyPrepData object

Details

This function launches a Shiny application that allows users to interactively explore the parameter space of the 'timer_transform' function. Users can adjust thresholds and normalization methods to see how these changes affect the transformation of flow cytometry data.

Value

Does not return a value; a Shiny app is launched in the default web browser.

Examples

```
## Not run:
  explore_timer_transform(prep_data, transformed_data)

## End(Not run)
```

plot_timer_gating *Plot Timer Gating Confirmation*

Description

Generates a plot of the negative control data with gating thresholds overlaid, allowing for visual confirmation of gating parameters used during the timer transformation process.

Usage

```
plot_timer_gating(prep, x)
```

Arguments

prep	A list containing file paths and variables, typically the output from prep_tocky. It must include: • neg: Character string specifying the negative control file name. • path: Character string specifying the directory path to data files.
x	A TockyPrepData object resulting from timer_transform, containing processed data and normalization parameters, including gating thresholds.

Details

This function reads the negative control data specified in prep, applies logarithmic transformation to the Timer Red and Timer Blue fluorescence channels, and plots Red_log versus Blue_log values. The gating thresholds extracted from the TockyPrepData object x are overlaid on the plot as vertical and horizontal lines. This allows users to visually confirm the gating thresholds applied during the data normalization process.

Value

This function generates a plot; it does not return a value.

Examples

```
## Not run:
# Assuming 'prep_data' is the output from 'prep_tocky' and 'tocky_data' is the TockyPrepData object
plot_timer_gating(prep = prep_data, x = tocky_data)

## 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(
  x,
  file = "PlotTocky",
  pseudocolour = TRUE,
  pdf = FALSE,
  output = "QC",
  n = 4,
  plot_mode = "Timer fluorescence",
  lower_quantile_cutoff = 0.01,
  select = FALSE,
  use_group = TRUE,
  group_order = NULL,
  interactive = TRUE,
```

```
  save = FALSE,
  samplefile = NULL,
  verbose = TRUE
)
```

Arguments

x	A TockyPrepData object returned by ‘timer_transform‘, which sample grouping has been defined by ‘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.
use_group	Logical indicating whether to group plots based on the ‘group‘ field in ‘sampledef‘.
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(
  x,
  sample_definition = NULL,
  output_dir = NULL,
  filename = "sampledef.csv",
```

```
  sep = ",",
  verbose = TRUE,
  interactive = FALSE
)
```

Arguments

x	A TockyPrepData object returned by ‘timer_transform‘.
sample_definition	(Optional) to use a data frame object as an annotation data for sample grouping. Defaul is ‘NULL‘.
output_dir	Character string specifying the directory to save the ‘sampledef.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 ‘"sampledef.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

An updated TockyPrepData with user-assigned groupings.

Examples

```
## Not run:
  # Assuming `x` is the output from `timer_transform`
  x <- sampledef(x, 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)
```

---

show,TockyPrepData-method

Show method for the TockyPrepData class

Description

Displays summary information for various slots of the TockyPrepData object. Includes details such as total number of cells, variable names, sample numbers, and group levels, providing a concise summary of the object.

Usage

```
## S4 method for signature 'TockyPrepData'
show(object)
```

Arguments

object	An object of the TockyPrepData class

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

Description

This function processes flow cytometry data by applying Timer thrsholding, normalization, and trigonometric transformation to the Blue and Red fluorescence data.

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,
  use_negative_control = 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'.
use_negative_control	Whether to use a negative control data to determine thresholds for Timer Blue and Timer Red, either by interactive mode or the quantile method. It is recommended to use the default 'TRUE'.

Value

The function returns a new TockyPrepData object containing:

- 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.
- sampledef: 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)
```

TockyPrepData-class *A class representing a TockyPrepData object for output of timer_transform*

Description

This class is designed to encapsulate and structure the output of the timer_transform function in the TockyPrep package.

Slots

Data A data.frame containing expression data.

cell_counts A data.frame containing counts of cells per sample.

sampledef A list including annotation data for sample grouping.

timer_fluorescence A list containing channel names for fluorescence timer data.

input A list of parameters used for creating TockyPrepData object.

normalization_parameters A list of parameters used for data normalization.

Tocky A list containing other Tocky-specific analysis data.

Index