Package: quicR (via r-universe)

February 12, 2025

Title RT-QuIC Data Formatting and Analysis

Version 2.1.0

Description Designed for the curation and analysis of data generated from real-time quaking-induced conversion (RT-QuIC) assays first described by Atarashi et al. (2011) <doi:10.1038/nm.2294>. 'quicR' calculates useful metrics such as maxpoint ratio: Rowden et al. (2023) <doi:10.1099/vir.0.069906-0>; time-to-threshold: Shi et al. (2013) <doi:10.1186/2051-5960-1-44>; and maximum slope. Integration with the output from plate readers allows for seamless input of raw data into the R environment.

Imports dplyr, ggplot2, janitor, openxlsx, purrr, readxl, reshape2, slider, stats, stringr, tidyr

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests testthat (>= 3.1.0)

Config/testthat/edition 3

Config/pak/sysreqs libicu-dev

Repository https://gage1145.r-universe.dev

RemoteUrl https://github.com/gage1145/quicr

RemoteRef HEAD

RemoteSha 47e57c91349b80ecde09b6235c6f3f1ee2f2e71b

Contents

add_reps 2
BMG_format 3
calculate_metrics 4
calculate_MPR 5

calculate_MS 6
calculate_threshold 6
calculate_TtT 7
convert_tables 8
get_meta 9
get_real 9
get_sample_locations 10
get_wells 11
normalize_RFU 12
organize_tables 13
plate_view 13
plot_metrics 14
separate_raw 15
transpose_real 16

Index **17**

`add_reps` *Add replicates*

Description

Adds replicate information to the sample IDs. Well IDs should be formatted like so: A4, B9, H11, J24

Usage

```
add_reps(df, sep = "_")
```

Arguments

`df`	A dataframe containing two columns for well IDs and Sample IDs
`sep`	a character string to separate the terms.

Value

A dataframe with replicate numbers pasted to the Sample IDs

`BMG_format` *Format Table for BMG Sample ID Import*

Description

BMG_format accepts a plate layout .CSV file and formats the Sample IDs into a format which can be easily imported into the BMG control software.

Usage

```
BMG_format(
  file,
  save_path = "",
  save_name = "formatted.txt",
  write_file = FALSE
)
```

Arguments

`file`	A .CSV file containing the plate layout of Sample IDs.
`save_path`	The path to the directory that you want the file saved.
`save_name`	The name of the output file. Should have the ".txt" extension.
`write_file`	Logical. If true, function will write a .txt file; otherwise it will return a character vector.

Value

A text file containing information for import into the BMG control software.

Examples

```
layout_file <- system.file(
  "extdata/BMG_formatting",
  file = "plate_layout.csv",
  package = "quicR"
)
BMG_format(layout_file)
```

`calculate_metrics` *Generate a dataframe with calculated metrics.*

Description

Uses functions from the "calculate" family of quicR functions to generate an analyzed dataframe.

Usage

```
calculate_metrics(
  data,
  meta,
  metrics = c("MPR", "MS", "TtT", "RAF"),
  transpose = FALSE,
  normalize = FALSE,
  start_col = 3L,
  MS_window = 3L,
  threshold = 2
)
```

Arguments

`data`	A dataframe containing the raw RT-QuIC data.
`meta`	A dataframe containing sample metadata. Should include at least the "Sample IDs" column.
`metrics`	An array containing the metrics which should be calculated.
`transpose`	Logical; should the raw data be transposed before performing the calculations?
`normalize`	Logical; should the raw data be normalized before performing the calculations?
`start_col`	Integer; column number denoting where the numeric data begins.
`MS_window`	Integer; width of the window applied in the calculation of max slope.
`threshold`	Float; the threshold applied to the calculation of time-to-threshold.

Value

A dataframe of calculated metrics.

Examples

```
file <- system.file(
  "extdata/input_files",
  file = "test4.xlsx",
  package = "quicR"
)
```

```
data <- quicR::get_real(file)[[1]] |>
  quicR::normalize_RFU()

meta <- quicR::organize_tables(file) |>
  quicR::convert_tables()

calculate_metrics(data, meta)
```

`calculate_MPR` *Calculate the Maxpoint Ratio*

Description

Maxpoint ratio is defined as the maximum relative fluorescence divided by the background fluorescence.

Usage

```
calculate_MPR(data, start_col = 3, data_is_norm = TRUE)
```

Arguments

`data`	A dataframe containing the real-time fluorescence data.
`start_col`	Integer, the column at which the background fluorescence should be read.
`data_is_norm`	Logical, if the data has not been normalized, will make a call to normalize_RFU.

Value

A vector containing MPR values.

Examples

```
# This test takes >5 sec

file <- system.file(
  "extdata/input_files",
  file = "test.xlsx",
  package = "quicR"
)
df_ <- quicR::get_real(file)[[1]]
print(calculate_MPR(df_))
```

calculate_MS *Calculate Maximum Slope*

Description

Uses a sliding window to calculate the slope of real-time reads.

Usage

```
calculate_MS(data, window = 3, data_is_norm = TRUE)
```

Arguments

data	A dataframe containing real-time reads. It is recommended to use a dataframe made from normalize_RFU.
window	Integer designating how wide you want the sliding window to be for calculating the moving average slope.
data_is_norm	Logical; if FALSE, will make a call to normalize_RFU.

Value

A dataframe containing the real-time slope values as ΔRFU/sec.

Examples

```
# This test takes >5 sec

file <- system.file(
  "extdata/input_files",
  file = "rt_data.csv",
  package = "quicR"
)
df_ <- read.csv(file, check.names = FALSE)
calculate_MS(df_)
```

calculate_threshold *Calculate a Threshold for Rate Determination*

Description

Calculates a threshold for determining time-to-threshold and rate of amyloid formation.

Usage

```
calculate_threshold(
  data,
  background_cycle = 2,
  method = list("stdev", "none"),
  multiplier = 1
)
```

Arguments

data	A dataframe output from get_real.
background_cycle	Integer; the cycle used for background fluorescence.
method	Method for determining threshold; default is "stdev".
multiplier	For some methods, will add a multiplier for more conservative thresholds.

Value

A float value.

Examples

```
file <- system.file(
  "extdata/input_files",
  file = "test2.xlsx",
  package = "quicR"
)
threshold <- get_real(file)[[1]] |>
  calculate_threshold(multiplier = 10)
```

---

calculate_TtT *Calculate Time to Threshold*

Description

Calculates the time required to reach a defined threshold.

Usage

```
calculate_TtT(data, threshold, start_col = 3)
```

Arguments

data	A dataframe containing real-time RT-QuIC data.
threshold	A numeric value defining the threshold.
start_col	The column containing the starting position of the real-time data.

Value

A vector containing the times to threshold

Examples

```
# This test takes >5 sec

file <- system.file(
  "extdata/input_files",
  file = "test2.xlsx",
  package = "quicR"
)
df_ <- get_real(file)[[1]] |>
  quicR::transpose_real() |>
  quicR::normalize_RFU(transposed = TRUE)
calculate_TtT(df_, threshold = 2)
```

---

convert_tables *Convert tables into a single column in a dataframe.*

Description

Accepts a table or matrix or a list of tables and matrices and converts them into dataframe columns.

Usage

```
convert_tables(tab, na_omit = TRUE)
```

Arguments

tab	A table/matrix or a list of tables/matrices.
na_omit	Logical; if true, will remove rows with NA.

Value

A dataframe column.

Examples

```
file <- system.file(
  "extdata/input_files",
  file = "test.xlsx",
  package = "quicR"
)
tabs <- organize_tables(file)
convert_tables(tabs)
```

get_meta *Retrieve the BMG metadata*

Description

Takes the Excel file exported from MARS and compiles the metadata in the header.

Usage

```
get_meta(file)
```

Arguments

file	The Excel file exported from MARS.

Value

A dataframe containing the Meta_ID and Meta_info

Examples

```
file <- system.file(
  "extdata/input_files",
  file = "test.xlsx",
  package = "quicR"
)
get_meta(file)
```

get_real *Get Real-Time RT-QuIC Fluorescence Data*

Description

Accepts an Excel file or a dataframe of real-time RT-QuIC data.

Usage

```
get_real(data, ordered = FALSE)
```

Arguments

data	Either an Excel file or a dataframe.
ordered	Logical, if true, will organize the columns by sample ID rather than by well.

Value

A list of dataframes containing the formatted real-time data.

Examples

```
file <- system.file(
  "extdata/input_files",
  file = "test.xlsx",
  package = "quicR"
)
get_real(file)
```

---

get_sample_locations *Get the well locations of the samples used in the RT-QuIC run.*

Description

Returns a dataframe with the sample IDs and well IDs used in the plate.

Usage

```
get_sample_locations(
  file,
  tab_name = "Sample IDs",
  dilution_bool = FALSE,
  dilution_fun = function(x) 1 * x,
  sep = "\n",
  plate = 96
)
```

Arguments

`file`	Excel file exported from MARS
`tab_name`	Table name containing the sample IDs.
`dilution_bool`	Logical; is there a table containing dilution factors? If so, will add a newline and the log of the dilution factor to the ID column.
`dilution_fun`	A function for transforming the dilution factor.
`sep`	A string used to separate the sample ID and dilution factor.
`plate`	Integer; either 96 or 384 to denote microplate type.

Value

A vector containing well IDs.

Examples

```
file <- system.file(
  "extdata/input_files",
  file = "test.xlsx",
  package = "quicR"
)
get_sample_locations(file)
```

`get_wells` *Get the Wells Used in the RT-QuIC Run.*

Description

Returns the well IDs used in the plate.

Usage

```
get_wells(file)
```

Arguments

`file`	Excel file exported from MARS

Value

A vector containing well IDs.

Examples

```
file <- system.file(
  "extdata/input_files",
  file = "test.xlsx",
  package = "quicR"
)
get_wells(file)
```

`normalize_RFU` *Normalize Fluorescence*

Description

Normalizes the real-time RT-QuIC data against the background fluorescence of a defined cycle. All cycles are divided by the fluorescent value of the defined cycle.

Usage

```
normalize_RFU(data, bg_cycle = 4, transposed = FALSE)
```

Arguments

`data`	A dataframe generated from get_real.
`bg_cycle`	The cycle used for background fluorescence
`transposed`	Logical, TRUE if cycle values are shown as column names.

Value

A dataframe containing real-time normalized fluorescence values.

Examples

```
# This test takes >5 sec

file <- system.file(
  "extdata/input_files",
  file = "test2.xlsx",
  package = "quicR"
)
df_ <- get_real(file)[[1]]

# Export the tables in the first sheet of the file.
dic <- quicR::organize_tables(file)

# Normalize the raw data against the background reading.
normalize_RFU(df_)
```

organize_tables *Organize MARS Tables*

Description

Extracts the tables from the microplate view sheet in the MARS Excel file and adds each table to a list.

Usage

```
organize_tables(file, plate = 96)
```

Arguments

file	An Excel file exported from MARS.
plate	Integer either 96 or 384 to denote microplate type.

Value

A list containing tibbles.

Examples

```
file <- system.file(
  "extdata/input_files",
  file = "test.xlsx",
  package = "quicR"
)
organize_tables(file)
```

plate_view *Real-Time Plate View*

Description

Converts the real-time data into a ggplot figure. The layout is either 8x12 or 16x24 for 96- and 384-well plates, respectively.

Usage

```
plate_view(df, meta, plate = 96)
```

Arguments

df	Real-time dataframe
meta	Dataframe containing well IDs and Sample IDs to title each facet.
plate	Integer either 96 or 384 to denote microplate type.

Value

A ggplot object

Examples

```
# This test takes >5 sec

file <- system.file(
  "extdata/input_files",
  file = "test2.xlsx",
  package = "quicR"
)

# Get the real-time data.
df_ <- get_real(file, ordered = FALSE)[[1]] |>
  as.data.frame()

sample_locations <- get_sample_locations(
  file,
  dilution_bool = TRUE,
  dilution_fun = function(x) -log10(x)
)

plate_view(df_, sample_locations)
```

---

plot_metrics *Plot metrics generated from the "calculate" family of quicR functions.*

Description

Generates a faceted figure of boxplots.

Usage

```
plot_metrics(
  data,
  sample_col = "Sample IDs",
  fill = "Dilutions",
  dilution_bool = TRUE,
  nrow = 2,
  ncol = 2
)
```

Arguments

`data`	A dataframe containing the calculated metrics from the "calculate" family of quicR functions.
`sample_col`	The name of the column containing the sample IDs.
`fill`	The column containing the fill aesthetic. Usually the dilutions column.
`dilution_bool`	Logical; should dilution factors be included in the plot?
`nrow`	Integer; number of rows to output in the plot.
`ncol`	Integer; number of columns to output in the plot.

Value

A ggplot object

Examples

```
file <- system.file(
  "extdata/input_files",
  file = "test4.xlsx",
  package = "quicR"
)

data <- quicR::get_real(file)[[1]] |>
  quicR::normalize_RFU()

meta <- quicR::organize_tables(file) |>
  quicR::convert_tables()

calculate_metrics(data, meta) |>
  plot_metrics()
```

`separate_raw` *Separate Real-Time Data into separate dataframes.*

Description

If multiple real-time reads were exported from MARS, separate_raw will parse them out and separate them. It will also export to an Excel file with each real-time data having its own sheet.

Usage

```
separate_raw(file, num_rows, export_name)
```

Arguments

`file`	An Excel file exported from MARS.
`num_rows`	Number of rows in the header to ignore.
`export_name`	The name of the original file or an orignal name.

Value

An Excel file with separated raw real-time data.

`transpose_real` *Transpose Real-Time Data*

Description

Transposes the real-time data table exported by the BMG software. Accepts output from the function, "get_real".

Usage

```
transpose_real(data)
```

Arguments

`data`	A dataframe generated from get_real.

Value

A transposed dataframe containing real-time normalized fluorescence values.

Index

add_reps, 2

BMG_format, 3

calculate_metrics, 4
calculate_MPR, 5
calculate_MS, 6
calculate_threshold, 6
calculate_TtT, 7
convert_tables, 8

get_meta, 9
get_real, 9
get_sample_locations, 10
get_wells, 11

normalize_RFU, 12

organize_tables, 13

plate_view, 13
plot_metrics, 14

separate_raw, 15

transpose_real, 16