Package 'flippant'

April 9, 2024

Title Dithionite Scramblase Assay Analysis

Description The lipid scrambling activity of protein extracts and purified scramblases is often determined using a fluorescence-based assay involving many manual steps. flippant offers an integrated solution for the analysis and publication-grade graphical presentation of dithionite scramblase assays, as well as a platform for review, dissemination and extension of the strategies it employs. The package's name derives from a play on the fact that lipid scrambling is also sometimes referred to as 'flipping'.
The package is originally published as Cotton, R.J., Ploier, B., Goren, M.A., Menon, A.K., and Graumann, J. (2017). ``flippant–An R package for the automated analysis of fluorescence-based scramblase assays.'' BMC Bioinformatics 18, 146. <DOI:10.1186/s12859-017-1542-y>.

Version 1.5.5

Author Johannes Graumann [cre, aut], Richard Cotton [aut]

Maintainer Johannes Graumann <johannes.graumann@uni-marburg.de>

Depends R (>= 3.5.0), ggplot2 (>= 2.2.1)

Suggests bookdown, devtools, microbenchmark, multipanelfigure, pander, roxygen2, tools

Imports data.table (>= 1.11.4), magrittr (>= 1.5), methods, minpack.lm (>= 1.2-1), plyr (>= 1.8.4), pracma (>= 2.3.3), stringi (>= 1.2.3), utils, withr (>= 2.1.2)

License GPL (>= 3)

RoxygenNote 7.2.3

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2024-04-09 16:30:02 UTC

R topics documented:

extract_case_study_data 2

flippant 3
parse_felix_32_output 3
parse_felix_gx_output 4
parse_fluorimeter_output 4
parse_FluorS_Essence_3.8_output 6
parse_manual_output 7
scramblase_assay_input_template 8

Index **14**

`extract_case_study_data` *Extract the case study dataset*

Description

Extracts the data files used by the case study from the zip archive.

Usage

```
extract_case_study_data(exdir = ".", files = NULL)
```

Arguments

`exdir`	A string giving the path to the extraction directory. Passed to `unzip`.
`files`	A character vector of files to extract, or `NULL` to extract all files. Passed to `unzip`.

Value

A character vector of the extracted file paths is invisibly returned.

Author(s)

Richard Cotton

See Also

`unzip`

Examples

```
extract_case_study_data(tempfile("flippant"))
```

flippant *flippant*

Description

A package for the analysis of data provided by dithionite scramblase assays.

Author(s)

Johannes Graumann

parse_felix_32_output *parse_felix_32_output*

Description

Parse spectra from files provided by a QuantaMaster fluorimeter (Photon Technology International, Inc., Edison, New Jersey) using `Felix32 v1.20`

Usage

```
parse_felix_32_output(x = NULL)
```

Arguments

x	`character` vector resulting from `readLines` of the corresponding data file.

Details

A helper function to `parse_fluorimeter_output`.

Value

See `parse_fluorimeter_output`.

Author(s)

Johannes Graumann

See Also

`parse_fluorimeter_output`, `parse_felix_gx_output`, `parse_FluorS_Essence_3.8_output`, `parse_manual_output`

---

parse_felix_gx_output *parse_felix_gx_output*

Description

Parse spectra from files provided by a QuantaMaster fluorimeter (Photon Technology International, Inc., Edison, New Jersey) using FelixGX v4.1

Usage

```
parse_felix_gx_output(x = NULL)
```

Arguments

x	character vector resulting from readLines of the corresponding data file.

Details

A helper function to parse_fluorimeter_output.

Value

See parse_fluorimeter_output.

Author(s)

Johannes Graumann

See Also

parse_fluorimeter_output, parse_felix_32_output, parse_FluorS_Essence_3.8_output, parse_manual_output

---

parse_fluorimeter_output
parse_fluorimeter_output

Description

Parse fluorimeter spectra

Usage

```
parse_fluorimeter_output(
  spec_file = NULL,
  timepoint_of_measurement = NULL,
  n_averaging = 10,
  determine_zero_time = TRUE,
  adjust = TRUE,
 file_type = c("auto", "FelixGXv4.1.0.3096", "Felix32v1.20", "FluorSEssencev3.8",
    "manual")
)
```

Arguments

spec_file	Path to a '*.txt' file as a character object.
timepoint_of_measurement	A numeric indicating the time (in sec) at which fluorescence extrema are calculated (DEPENDENT ON adjust!).
n_averaging	A numeric indicating the number of data points used for extrema calculations.
determine_zero_time	A logical indicating whether (default) or not the timepoint of dithionite addition should be determined using **pracma**-derived functionality.
adjust	A logical indicating of whether (default) or not acquisition time should be reset to have 0 (zero) coincide with the addition of dithionite (see 'Details' section).
file_type	A string specifying whether or the file was created using Felix GX or Felix 32 or FluorS Essence v3.8 or is a "manual" tab delimited file.

Details

A function to read fluorimeter output directly. Intended as a helper function to scramblase activity determinations from dithionite assays.

The function is currently capable to deal with input derived from QuantaMaster instruments (Photon Technology International, Inc., Edison, New Jersey)running software versions FelixGX v4.1 (see parse_felix_gx_output), Felix32 v1.20 (see parse_felix_32_output) as well as Horiba fluorimeters (HORIBA Europe GmbH, Oberursel, Germany) using FluorS Essence v3.8. The format used in a given file is divined from the data structure and appropriate internal parsing functions are called.

If requested the time point of dithionite addition to a sample is determined using **pracma**-supplied methodology and the acquisition time reset accordingly (0 henceforth corresponds to the time of addition).

Value

A data frame with two columns:

Time.in.sec Numeric. Number of seconds since the start of experiment.

Fluorescence.Intensity Numeric. Intensity of fluorescence (relative scale, no official unit).

If `determine_zero_time` and/or `adjust` are set to `TRUE`, the return value will have an attribute `ZeroTimePoint` corresponding to the determined time point of dithionite addition (always `0` (zero) where `adjust == TRUE`).

For Felix GX, if the file contains the information, the return value will also have an attribute `WavelengthsInNanometres`, which contains the excitation and emission wavelengths.

Author(s)

Johannes Graumann

See Also

`scramblase_assay_input_validation`, `parse_felix_gx_output`, `parse_felix_32_output`, `parse_FluorS_Essence_3`
`parse_manual_output`

Examples

```
library(magrittr)
# Extract example data
analysis_dir <- file.path(tempdir(), "flippant-case-study")
fluor_file <- extract_case_study_data(analysis_dir, "wildtypeEx1_0.txt")
# Parse an exemplary file
parse_fluorimeter_output(fluor_file, timepoint_of_measurement = 350) %>%
  str()
```

`parse_FluorS_Essence_3.8_output`

parse_FluorS_Essence_3.8_output

Description

Parse spectra from files provided by a Horiba fluorimeter (HORIBA Europe GmbH, Oberursel, Germany) using `FluorS Essence v3.8`

Usage

```
parse_FluorS_Essence_3.8_output(x = NULL)
```

Arguments

x	`character` vector resulting from `readLines` of the corresponding data file.

Details

A helper function to `parse_fluorimeter_output`.

Value

See `parse_fluorimeter_output`.

Author(s)

Johannes Graumann

See Also

parse_fluorimeter_output, parse_felix_32_output, parse_felix_gx_output, parse_manual_output

---

parse_manual_output *parse_manual_output*

Description

Parse spectra from files provided in a manually assembled format.

Usage

```
parse_manual_output(x = NULL)
```

Arguments

x	character vector resulting from readLines of the corresponding data file.

Details

A helper function to parse_fluorimeter_output. The file in question is supposed to contain a tab-delimited table with the columns Time (sec) and Fluorescense Intensity.

Value

See parse_fluorimeter_output.

Author(s)

Johannes Graumann

See Also

parse_fluorimeter_output, parse_felix_gx_output, parse_felix_32_output

scramblase_assay_input_template *scramblase_assay_plot*

Description

Functions for the presentation and evaluaton of dithionite scramblase assays

Usage

```
scramblase_assay_input_template(
  path = "scramblase_assay_input_template.txt",
  input_directory = NULL,
  overwrite = FALSE
)

scramblase_assay_plot(
  x,
  scale_to = c("model", "data"),
  ppr_scale_factor = 0.65,
  force_through_origin = TRUE,
  generation_of_algorithm = c(2, 1),
  split_by_experiment = TRUE,
  r_bar = 88,
  sigma_r_bar = 28
)

scramblase_assay_stats(
  x,
  scale_to = c("model", "data"),
  ppr_scale_factor = 0.65,
  force_through_origin = TRUE,
  generation_of_algorithm = c(2, 1),
  split_by_experiment = TRUE,
  r_bar = 88,
  sigma_r_bar = 28
)

scramblase_assay_traces(
  x,
  ppr_scale_factor = 0.65,
  time_min_sec = NA_real_,
  time_max_sec = NA_real_,
  adjust = TRUE,
  timepoint_of_measurement = 400,
  n_averaging = 10,
  annotate_traces = FALSE
```

)

Arguments

path	character object giving the path of an **empty** template for a spreadsheet that can provide x.
input_directory	if not NULL, character object giving the path to a directory where spectrometer output resides for the prepopulation of the template spreadsheet.
overwrite	logical object allowing to overwrite existing template paths.
x	data.frame or path to a tab delimited file representing it (see "Details").
scale_to	Defines the source of ymax, defaulting to model. See "Details".
ppr_scale_factor	numeric object providing a scale factor to adjust internally calculated Protein per Phospholipid (mg/mmol) ratios (PPR; see "Details").
force_through_origin	logical indicating whether to force the fitted curve(s) to penetrate the origin (defaulting to TRUE). See "Details".
generation_of_algorithm	Either 2 or 1 (numeric; defaulting to 2). See "Details".
split_by_experiment	A single logical, indicating whether or not calculations and plots will treat experimental series from different experiments separately (TRUE, default) or whether data from all experiments included is used for a single calculation/plot per experimental series (FALSE). While the former emphasizes reproducibility, the latter likely produces a more reliable fit.
r_bar	A numeric, representing the average radius of the liposomes used in the assay. Only used in generatio_of_algorithm = 2 and defaulting to 88 (see Ploier et al. 2016 for details).
sigma_r_bar	A numeric, representing the standard deviationaverage of the radius distribution of the liposomes used in the assay. Only used in generatio_of_algorithm = 2 and defaulting to 28 (see Ploier et al. 2016 for details).
time_min_sec	A single numeric. If given, scramblase_assay_traces produces a time/x axis trimmed to this value (in seconds).
time_max_sec	A single numeric. If given, scramblase_assay_traces produces a time/x axis trimmed to this value (in seconds).
adjust	A single logical, indicating whether (default) or not spectral traces to be plotted are algorithmically aligned at the time point of dithionite addition.
timepoint_of_measurement	A numeric indicating the time (in sec) at which fluorescence extrema are calculated (DEPENDENT ON adjust!).
n_averaging	A numeric indicating the number of data points used for extrema calculations.
annotate_traces	A logical idicating whether fluorescence traces should be annotated.

Details

The data.frame accepted by the majority of the functions a an R object or path to a corresponding file (x) must have the following **mandatory** columns:

`Path`**:** Paths to existing and readable `ASCII` output files of a fluorimeter. See `parse_fluorimeter_output` for details and supported formats.

`Protein Reconstituted (mg)`**:** Self-explanatory. In the case of `scramblase_assay_traces` **ONLY** this may be abused by taking `character` values rather than the usually required `numeric`s. Handy when e.g. plotting traces for "Liposomes" and "Proteoliposomes", rather than defined PPRs.

Further (**facultative**) columns are:

`Fluorescence Assay Vol. w/o DT (ul)`**:** Volume of the fluorescence assay prior to addition of dithionite (defaulting to `2000`).

`Fluorescence Assay Vol. with DT (ul)`**:** Volume of the fluorescence assay after the addition dithionite (defaulting to `2040`).

`Lipid in Reconstitution (mmol)`**:** Self-explanatory. For the standard phospholipid experiment defaulting to `0.0045` (1 ml of a 4.5 mM solution).

`Timepoint of Measurement (s)`**:** The time to determine terminal fluorescence, calculated from the point when dithionite is added, in seconds, defaulting to `400`).

`Experiment`**:** Identifier for any given experiment. Used for `facet_wrap` during generation of `ggplot` output. All data with one `Experiment` identifier ends up on one plot/facet.

`Experimental Series`**:** Identifier for a given series/graph (e.g. `Extract` and `Depleted Extract`). Used by `color` during generation of `ggplot` output to differentiate lines in the same plot/facet.

Based on Goren et al. (2014) and Ploier et al. (2016) data is processed as follows (the majority of the processing is split off into the internal function `scramblase_assay_calculations`):

- Input is format checked and defaults are injected for facultative parameters/columns as appropriate (see input `data.frame` format above). The internal function `scramblase_assay_input_validation` supplies this functionality.
- Fluorescence spectra are parsed using `parse_fluorimeter_output`. This includes automated determination of when dithionite was added to the sample using **pracma**-supplied methodology and resetting the acquisition time accordingly (`0` henceforth corresponds to the time of addition).
- Pre-dithionite-addition `Baseline Fluorescence` is determined for each spectrum by averaging (`median`) over the 10 values preceding dithionite addition.
- Post-dithinonite-addition `Minimum Fluorescence` is determined for each spectrum by averaging (`median`) over the last ten datapoints ≤ 400 s (or `Timepoint of Measurement (s)`, see above).
- The `Minimum Fluorescence` is volume-corrected based on `Reaction Volume w/o DT (ul)` and `Reaction Volume with DT (ul)` (see above).
- For each spectrum/datapoint a measured `Fluorescence Reduction` is calculated as

$$1 - \left(\frac{\text{Minimum Fluorescence}}{\text{Baseline Fluorescence}} \right)$$

- A `Relative Fluorescence Reduction` is calculated in comparison to the liposomes-only/no-protein control).
- A `Protein per Phospholipid (mg/mmol)` ratio (`PPR`) is calculated. If `ppr_scale_factor` is not `NULL`, the value is scaled (divided) by that value to account for liposomes that remain inaccessible to reconstitution with scramblase molecules.
- Depending on `split_by_experiment`, data are `split` for parallel treatment using either `Experimental Series` (`split_by_experiment = TRUE`) or a combined `Experimental Series/Experiment` (`split_by_experiment = FALSE`) identifier (see above).
- A probability for a liposome holding ≥ 1 scramblase molecules is calculated using

$$\frac{y - y_0}{y_{\max} - y_0}$$

  where y is the `Relative Fluorescence Reduction` and y_0 is the `Relative Fluorescence Reduction` in an experiment without addition of protein extract. Depending on the `scale_to` parameter, $y_{\max}$ is either the maximal `Relative Fluorescence Reduction` in the series (`scale_to = "data"`) or derived from a mono-exponential fit to the data (`scale_to = "model"`). The latter (default) is a precaution for the case where the protein/phospholipid titration did not reach the plateau of the saturation curve.
- A monoexponential curve is fitted using `nlsLM`.

  If `generation_of_algorithm` is `1`, the underlying formula is derived from Goren et al. (2014) and data is fitted to either

$$p(\geq 1) = b \cdot (1 - e^{-\frac{\text{PPR}}{a}})$$

  (if `force_through_origin = TRUE`; default) or

$$p(\geq 1) = b - c \cdot e^{-\frac{\text{PPR}}{a}}$$

  (if `force_through_origin = FALSE`). The latter implies more degrees of freedom and occasionaly results in better fits to experimental data. Mechanistic implication, however, are unclear.

  If `generation_of_algorithm` is `2` (default), the more elaborate model put forth in Ploier et al. (2016) is employed, using either

$$p(\geq 1) = b \cdot (\frac{1}{\sqrt{1 + \sigma^2 \cdot a \cdot x}}) \cdot exp(\frac{-\bar{r}^2 \cdot a \cdot x}{1 + \sigma^2 \cdot a \cdot x})$$

  (if `force_through_origin = TRUE`; default) or

$$p(\geq 1) = b - c \cdot (\frac{1}{\sqrt{1 + \sigma^2 \cdot a \cdot x}}) \cdot exp(\frac{-\bar{r}^2 \cdot a \cdot x}{1 + \sigma^2 \cdot a \cdot x})$$

  (if `force_through_origin = FALSE`).
- Data `split` apart above are recombined and a `ggplot` object is assembled with the following layers:
  - Lines (`geom_line`) representing the monoexponential fit(s). `color` is used to differentiate `Experimental Series`.

- If generation_of_algorithm is 1, segments (geom_segment) representing the PPR at which the fit constant a is equal to PPR. This τ value has the implication that at this PPR all vesicles on average have one scramblase and 63% have one or more (i.e. are active). color is used to differentiate Experimental Series. Where generation_of_algorithm is 2, interpretation of a is less obvious and this layer is thus ommited in the plot.
- Points (geom_point) representing the corresponding datapoints. color is used to differentiate Experimental Series.
- Plots are finally facet_wraped by Experiment (if split_by_experiment = TRUE) and labels adjusted cosmetically.

Value

scramblase_assay_traces and scramblase_assay_plot return ggplot objects representing the raw fluorescence traces and a complete PPR plot, respectively. scramblase_assay_input_template generates a tab-delimited ASCII file in the file system and invisibly returns the path name. scramblase_assay_stats assembles (and prints) assay statistics as a data.frame.

Author(s)

Johannes Graumann

References

Menon, I., Huber, T., Sanyal, S., Banerjee, S., Barre, P., Canis, S., Warren, J.D., Hwa, J., Sakmar, T.P., and Menon, A.K. (2011) <DOI:10.1016/j.cub.2010.12.031>

Goren, M.A., Morizumi, T., Menon, I., Joseph, J.S., Dittman, J.S., Cherezov, V., Stevens, R.C., Ernst, O.P., and Menon, A.K. (2014) <DOI:10.1038/ncomms6115>

Ploier, B., Caro, L.N., Morizumi, T., Pandey, K., Pearring, J.N., Goren, M.A., Finnemann, S.C., Graumann, J., Arshavsky, V.Y., Dittman, J.S., Ernst, O.P., Menon, A.K. (2016). <DOI:10.1038/ncomms12832>

See Also

parse_fluorimeter_output nlsLM

Examples

```
library(magrittr)
library(ggplot2)
# Extract example data
analysis_dir <- file.path(tempdir(), "flippant-case-study")
extract_case_study_data(analysis_dir)
template_file <- file.path(analysis_dir, "inputTable.txt")
# Plot the spectral traces
scramblase_assay_traces(
  template_file,
  time_max_sec = 350,
  timepoint_of_measurement = 350)
# Plot the PPR plot(s) faceting by experiment
scramblase_assay_plot(template_file)
# Generate tabular results
```

```
scramblase_assay_stats(template_file)
# Plot the PPR plot(s) forgoing faceting by experiment
scramblase_assay_plot(template_file, split_by_experiment = FALSE)
# Generate tabular results
scramblase_assay_stats(template_file, split_by_experiment = FALSE)
```

Index

∗ **IO**
  parse_felix_32_output, 3
  parse_felix_gx_output, 4
  parse_fluorimeter_output, 4
  parse_FluorS_Essence_3.8_output, 6
  parse_manual_output, 7
∗ **file**
  parse_felix_32_output, 3
  parse_felix_gx_output, 4
  parse_fluorimeter_output, 4
  parse_FluorS_Essence_3.8_output, 6
  parse_manual_output, 7
∗ **manip**
  parse_felix_32_output, 3
  parse_felix_gx_output, 4
  parse_fluorimeter_output, 4
  parse_FluorS_Essence_3.8_output, 6
  parse_manual_output, 7
∗ **package**
  flippant, 3

character, *3–7*, *9*, *10*

data.frame, *9*, *10*, *12*

extract_case_study_data, 2

facet_wrap, *10*, *12*
flippant, 3

geom_line, *11*
geom_point, *12*
geom_segment, *12*
ggplot, *10–12*

logical, *5*, *9*

median, *10*

nlsLM, *11*, *12*
numeric, *5*, *9*, *10*

parse_felix_32_output, 3, *4–7*
parse_felix_gx_output, *3*, 4, *5–7*
parse_fluorimeter_output, *3*, *4*, 4, *6*, *7*, *10*, *12*
parse_FluorS_Essence_3.8_output, *3*, *4*, *6*, 6
parse_manual_output, *3*, *4*, *6*, *7*, 7

readLines, *3*, *4*, *6*, *7*

scramblase_assay_calculations *(*scramblase_assay_input_template*)*, 8
scramblase_assay_input_template, 8
scramblase_assay_plot *(*scramblase_assay_input_template*)*, 8
scramblase_assay_stats *(*scramblase_assay_input_template*)*, 8
scramblase_assay_traces, *9*, *10*
scramblase_assay_traces *(*scramblase_assay_input_template*)*, 8
split, *11*

unzip, *2*