

Package: reluxr (via r-universe)

October 15, 2024

Title Deconvolute Luminescence Readings on Bacterial Culture Plates

Version 0.1.1

Description Implements the deconvolution algorithm developed in Mauri, Vecchione, and Fritz (2019) which enables deconvolution of luminescence readings for experimental culture plates. {reluxr} provides functions for calculating the 'best' deconvolution matrix from a calibration plate, and enables usage of this calibration matrix (or one calculated previously) to adjust experimental values from a plate reader.

License MIT + file LICENSE

Suggests covr, readxl, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

Depends R (>= 4.1.0)

Imports cli, rlang, scales, dplyr, tidyr, purrr, janitor, ggplot2, stringr

URL <https://github.com/bradyajohnston/reluxr>

BugReports <https://github.com/bradyajohnston/reluxr/issues>

Repository <https://bradyajohnston.r-universe.dev>

RemoteUrl <https://github.com/bradyajohnston/reluxr>

RemoteRef HEAD

RemoteSha dcc567456a372cb018856062110dc914583a67ce

Contents

<code>.deconvolute_multi_frame_matrix</code>	2
<code>.decon_vec</code>	3
<code>.mat_calc_bleed</code>	3
<code>.mat_calc_deconvolution</code>	4
<code>.multi_frame_matrix_from_df</code>	4
<code>.multi_frame_matrix_to_vec</code>	5
<code>df_arrange</code>	6
<code>is_well_id</code>	6
<code>plate_read_tecan</code>	7
<code>rl_adjust_plate</code>	7
<code>rl_calc_decon_matrix</code>	9
<code>rl_mat_decon_best</code>	10
<code>rl_plot_plate</code>	10
Index	12

`.deconvolute_multi_frame_matrix`

Deconvolute a Multi-Frame Matrix

Description

Deconvolute a Multi-Frame Matrix

Usage

```
.deconvolute_multi_frame_matrix(mat_frames, mat_decon)
```

Arguments

<code>mat_frames</code>	A multi-frame matrix, with each time point a row, and each column a well.
<code>mat_decon</code>	A deconvolution matrix created through <code>.mat_calc_deconvolution()</code> .

Value

A deconvoluted multi-frame matrix.

.decon_vec *Deconvolute a Single Vector*

Description

Deconvolute a Single Vector

Usage

```
.decon_vec(vec, mat_decon)
```

Arguments

vec Numeric vector, representing the wells.
mat_decon Deconvolution matrix created through .mat_calc_deconvolution().

Value

a numeric vector, the same length as vec.

.mat_calc_bleed *Calculate Bleed Through Matrix*

Description

Expand and calculate a bleed-through matrix from the given matrix around the given reference well.

Usage

```
.mat_calc_bleed(  
  mat,  
  ref_row,  
  ref_col,  
  b_noise = 10,  
  relative = TRUE,  
  .f = mean  
)
```

Arguments

mat A matrix to expand and calculate relative bleed-through based on a reference well.
ref_row Reference well row number.
ref_col Reference well column number.
b_noise Calculated background noise to fill the bleed-through matrix.
relative TRUE / FALSE whether to return relative values.
.f Function to apply to the expanded matrix, defaults to mean.

Value

a matrix, that is a bleed-through matrix calculated from the given matrix.

```
.mat_calc_deconvolution
```

Make Deconvolution Matrix From Bleed-Through Matrix

Description

Make Deconvolution Matrix From Bleed-Through Matrix

Usage

```
.mat_calc_deconvolution(mat)
```

Arguments

mat Bleed-through matrix from `.mat_calc_bleed()`

Value

a matrix, ready for deconvolution.

```
.multi_frame_matrix_from_df
```

Create a multi-frame matrix from a dataframe.

Description

Creates a matrix with each row being a time point in a multi-frame experiment, and each column is a single well. The columns are concatenated by row (ordered A1, A2, ... B1, B2, ...).

Usage

```
.multi_frame_matrix_from_df(  
  data,  
  value,  
  time = "time",  
  well = "well",  
  arrange = FALSE  
)
```

Arguments

<code>data</code>	A dataframe with columns for the value, time and well ID.
<code>value</code>	Name of the column which contains the values for the matrix.
<code>time</code>	Name of the column which defines the time points for the frames (rows) of the matrix.
<code>well</code>	Name of the column which contains the well IDs of the samples.
<code>arrange</code>	Logical, whether to return a dataframe arranged by time and well.

Value

a matrix, with

`.multi_frame_matrix_to_vec`

Convert a Multi-Frame Matrix to a Vector

Description

Takes a multi-frame matrix and converts to a vector, with each frame being concatenated end-to-end. Each row / frame of the matrix is a time point, with each column representing a well (ordered A1, A2, ... B1, B2, ...).

Usage

```
.multi_frame_matrix_to_vec(mat, rowwise = TRUE)
```

Arguments

<code>mat</code>	A multi-frame matrix.
<code>rowwise</code>	Logical, whether to concatenate the rows (default, TRUE) or the columns.

Value

A vector, with values concatenated from the matrix.

df_arrange	<i>Reorder a DataFrame</i>
------------	----------------------------

Description

Reorders a dataframe by the time points, then by the rows then by the columns.

Usage

```
df_arrange(data, time = "time", well = "well")
```

Arguments

data	A dataframe with a time column and a well column.
time	The name of the column with the time points.
well	The name of the column with the well ID information.

Value

The input dataframe reordered.

is_well_id	<i>Logical test for well ID format.</i>
------------	---

Description

Logical test for well ID format.

Usage

```
is_well_id(x)
```

Arguments

x	A string vector.
---	------------------

Value

A logical vector.

Examples

```
is_well_id(c("a12", "a2", "a02", "foo1"))
```

plate_read_tecan *Read the output of Tecan Plate Readers*

Description

Read the output of Tecan Plate Readers

Usage

```
plate_read_tecan(file, temp = FALSE)
```

Arguments

file File path to the .xlsx or .csv file.
temp Logical, whether to include the temperature column.

Value

a tibble::tibble() of the values from the file.

Examples

```
f1 <- system.file(  
  "extdata",  
  "calibrate_tecan",  
  "calTecan1.xlsx",  
  package = "reluxr"  
)  
  
dat <- plate_read_tecan(f1)  
  
dat
```

r1_adjust_plate *Adjust Experimental Luminescent Data*

Description

Using a deconvolution matrix, created through `r1_calc_decon_matrix()`, adjust the values in the `col_value` column to take into account bleed-through from surrounding wells.

Usage

```
r1_adjust_plate(data, value, mat_decon, time = "time", well = "well")
```

Arguments

data	A data frame that contains the experimental data.
value	The name of the column containing the values (i.e. 'lum').
mat_decon	A deconvolution matrix created through <code>rl_calc_decon_matrix()</code>
time	The name of the column containing the time values (i.e. 'time')
well	Name of the column with the well ID information.

Value

A dataframe with the specified column having been deconvoluted, using the supplied deconvolution matrix.

Examples

```
f1 <- system.file(
  "extdata",
  "calibrate_tecan",
  "calTecan1.xlsx",
  package = "reluxr"
)

dat <- plate_read_tecan(f1)

mat_d_best <- dat |>
  dplyr::filter(signal != "OD600") |>
  dplyr::filter(time_s > 500) |>
  rl_calc_decon_matrix(value, time_s, ref_well = "E05", b_noise = 30)

dat |>
  dplyr::summarise(value = mean(value), .by = well) |>
  rl_plot_plate(value, trans = log10) +
  ggplot2::scale_fill_viridis_c(
    limits = c(1, NA)
  )

dat |>
  dplyr::filter(signal == "LUMI") |>
  rl_adjust_plate(value, mat_d_best, time = time_s) |>
  dplyr::summarise(value = mean(value), .by = well) |>
  rl_plot_plate(value, trans = log10) +
  ggplot2::scale_fill_viridis_c(
    limits = c(1, NA)
  )
```

rl_calc_decon_matrix *Calculate a Deconvolution Matrix*

Description

Use the data from a calibration plate, where the plate is empty except for a single well with a luminescent signal, to create a deconvolution matrix that can be used to adjust other experimental results.

Usage

```
rl_calc_decon_matrix(  
  data,  
  value,  
  b_noise,  
  time = "time",  
  ref_well = "I05",  
  well = "well"  
)
```

Arguments

data	A data frame that contains the data of the calibration plate.
value	Name of the column containing the luminescent values.
b_noise	The value of the background noise, which is the average signal for the background wells that are far away from the reference well.
time	Name of the column with the time values.
ref_well	The well ID of the reference well (i.e. 'E05', 'I12')
well	Name of the column with the well ID values.

Details

The deconvolution matrix will be unique for each plate type and plate-reader, so a matrix should be calculated for each combination of plate and plate reader, but once this is calculated, it can be re-used to adjust future experimental results.

Value

a deconvolution matrix, for use in `rl_adjust_plate()`

Examples

```
f1 <- system.file(  
  "extdata",  
  "calibrate_tecan",  
  "calTecan1.xlsx",
```

```

  package = "reluxr"
)

dat <- plate_read_tecan(fl)

dat

mat_d <- dat |>
  dplyr::filter(signal != "OD600") |>
  dplyr::filter(time_s > 500) |>
  rl_calc_decon_matrix(value, time_s, ref_well = "E05", b_noise = 30)

image(log10(mat_d))

```

rl_mat_decon_best *Calculate the Optimal Deconvolution Matrix*

Description

Calculate the Optimal Deconvolution Matrix

Usage

```
rl_mat_decon_best(mat, ref_row, ref_col, b_noise = 20)
```

Arguments

mat	A multi-frame matrix, where each row is a frame and each column is a well.
ref_row	Row number for the reference well.
ref_col	Column number for the reference well.
b_noise	Value for the background noise.

Value

a matrix, the optimised deconvolution matrix.

rl_plot_plate *Plot a Plate-Layout of the Value Column*

Description

Plot a Plate-Layout of the Value Column

Usage

```
rl_plot_plate(data, value, well = "well", trans = log10)
```

Arguments

data	Dataframe with the value column and a column specifying the well ID.
value	Name of the column containing the value information to be displayed.
well	Name of the column with the well ID information for formatting the plate layout.
trans	Name of the transformation to apply to the data. Defaults to log10.

Value

a `ggplot2::ggplot()` plot.

Examples

```
f1 <- system.file(
  "extdata",
  "calibrate_tecan",
  "calTecan1.xlsx",
  package = "reluxr"
)

dat <- plate_read_tecan(f1)

mat_d_best <- dat |>
  dplyr::filter(signal == "LUMI") |>
  dplyr::filter(time_s > 500) |>
  rl_calc_decon_matrix("value", "time_s", ref_well = "E05", b_noise = 30)

dat |>
  dplyr::filter(signal == "LUMI") |>
  dplyr::filter(time_s > 500) |>
  rl_adjust_plate(value, mat_d_best, time = time_s) |>
  dplyr::summarise(value = mean(value), .by = well) |>
  rl_plot_plate(value, trans = log10) +
  ggplot2::scale_fill_viridis_c(
    limits = c(1, NA)
  )
)
```

Index

.decon_vec, 3
.deconvolute_multi_frame_matrix, 2
.mat_calc_bleed, 3
.mat_calc_deconvolution, 4
.multi_frame_matrix_from_df, 4
.multi_frame_matrix_to_vec, 5

df_arrange, 6

is_well_id, 6

plate_read_tecan, 7

rl_adjust_plate, 7
rl_calc_decon_matrix, 9
rl_mat_decon_best, 10
rl_plot_plate, 10