

Package ‘bioassays’

May 29, 2020

Type Package

Title Summarising Multi Well Plate Cellular Assay

Version 0.1.0

Description The goal is to help users to analyse data from multi wells with minimum effort. Using these functions several plates can be analyzed automatically.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports dplyr, ggplot2, stats, magrittr, nplr, reshape2, rlang

RoxygenNote 7.1.0

Depends R (>= 4.00)

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Anwar Azad Palakkan [aut, cre],
Jamie Davies [aut]

Maintainer Anwar Azad Palakkan <bioanwar@gmail.com>

Repository CRAN

Date/Publication 2020-05-29 13:20:06 UTC

R topics documented:

bioassays	2
data2plateformat	3
data_DF1	4
dfsummary	4
estimate	6
extract_filename	7
heatplate	8
matrix96	9

metafile384	10
metafile96	11
plate2df	11
plate_metadata	12
pvalue	13
rawdata24	14
rawdata384	15
rawdata96	15
reduceblank	16
rmodd_summary	17
Index	19

bioassays

Bioassays: A Package for Analyzing Multi Well Plate Bioassay's.

Description

The bioassays package provides three categories of important functions for extracting, formatting, plotting and analyzing.

Extract functions. The function that help to extract information from file name is [extract_filename](#)

NA

Format functions. The function to format data are [data2plateformat](#),[plate2df](#),[matrix96](#),[plate_metadata](#)

NA

Plot functions. The function to plot is [heatplate](#)

NA

Analysis functions. The function to analyse data are [reduceblank](#),[estimate](#),[dfsummary](#),[pvalue](#)

NA

data2plateformat *Renaming column and Row of Multiwell Data to Match Plate Format*

Description

Convert the data (example: readings from mutli well plate) to appropriate plate format by renaming column and rownames.

Usage

```
data2plateformat(data, platetype = 96)
```

Arguments

data	Matrix data to be formatted.
platetype	Plate from which the data is coming. It can take 6, 12, 24, 96 and 384 values to represent the corresponding multi well plate.

Details

This function will label the columns and rows correctly to match the plate format, and discard the extras. For example, if the 'data' is coming from a '96' well plate ('platetype'), the function will rename rows as A to H and columns as 1 to 12. Extra columns and rows of 'data' is discarded.

Value

A data frame with columns and rows matching (label and numbers) the mutli well plate format.

Author(s)

A.A Palakkan

Examples

```
## loading data
data(rawdata24,rawdata96,rawdata384)

## eg:1 spectrophotometer reading from 24 well plate
data2plateformat(rawdata24, platetype = 24)

## eg:2 spectrophotometer reading from 96 well plate
data2plateformat(rawdata96, platetype = 96)

## eg:3 spectrophotometer reading from 384 well plate
data2plateformat(rawdata384, platetype = 384)
```

data_DF1	<i>Data of 96 well plate.</i>
----------	-------------------------------

Description

A complete dataset containing both metadata and spectrophotometer reading.

Usage

```
data_DF1
```

Format

A data frame with 96 rows and 10 variables

row Row number of multi well well plate.

col Column number of multi well plate.

position Well position address of multi well plate.

value Spectrophotometer reading (OD).

id Type of sample. 'STD' represent standards and 'sample' represent samples.

type Type of sample. 'STD1', 'STD2', 'STD3' etc represent different standards. 'S1', 'S2', 'S3' represent different samples.

dilution dilution of samples used for the assay.

concentration Concentration of respective standards.

compound Compound used for the assay.

blankminus Blank reduced OD (value - mean(blank))

Source

User generated dataframe of the 96 well plate.

dfsummary	<i>Summarize a Dataframe After Grouping Samples</i>
-----------	---

Description

This function summarize the dataframe (based on a column). It has additional controls to group samples and to omit variables not needed.

Usage

```
dfsummary(dataframe, y, grp_vector, rm_vector, nickname, rm="FALSE", param)
```

Arguments

dataframe	data in dataframe format
y	column name whose values has to be summarized (column elements need to be numeric)
grp_vector	a character vector of column names whose order indicate the order of grouping
rm_vector	a character vector of items that need to be omitted before summarizing
nickname	label name for the entries in output dataframe
rm	rm = "FALSE" if outliers not to be removed. rm = "TRUE" If outliers to be removed
param	a vector of parameters for more stringent outlier removal. param has to be entered in the format c(strict, cutoff, n). For details please refer rmodd_summary ..

Details

This function first remove 'rm_vector' elements from the 'dataframe'. Samples are grouped (each level of a 'grp_vector' element as separate group) and sorted (based on 'grp_vector' elements order). column 'y' is then summarized for each group (please refer [rmodd_summary](#): for details).

Value

A dataframe. First columns are named as grp_vector elements. Followed by a 'label' column (element is 'nickname'). This 'label' column will be useful when analyzing multiple plates. Summary statistics of 'y' appear as columns: N (number of samples/group), Mean (average/group), SD (standard deviation/group) and CV (percentage cv/group).

Author(s)

A.A Palakkan

Examples

```
## loading data
data(metafile384, rawdata384)
rawdata<-plate2df(data2plateformat(rawdata384,platetype = 384))
data_DF2<- dplyr::inner_join(rawdata,metafile384,by=c("row","col","position"))

## eg:1 summarising the 'value' after grouping samples and omitting blanks.
# grouping order cell, compound, concentration and type.

result2 <- dfsummary(data_DF2,y = "value",
  grp_vector = c("cell","compound","concentration","type"),
  rm_vector = c("blank1","blank2","blank3","blank4"),
  nickname = "384well",
  rm = "FALSE",param = c(strict="FALSE",cutoff=40,n=12))
```

 estimate

Estimating Variable Based on Standard Curve

Description

This function will estimate the unknown variable (example: concentration) based on a standard curve.

Usage

```
estimate (data, colname = "blankminus", fitformula = fiteq, method = "linear/nplr")
```

Arguments

data	data in dataframe format
colname	column name whose values has to be estimated
fitformula	formula used for fitting standard curve
method	method = "linear" if standard curve is linear in nature. method = "nplr" if standard curve is nonparametric logistic curve.

Details

For linear standard curve 'fitformula' need to generated using [lm](#). For nonparametric logistic curve 'fitformula' need to generated using [nplr](#).

Value

A dataframe with estimated values added to right as a new column "estimated".

Author(s)

A.A Palakkan

Examples

```
## loading data
data(data_DF1)

## Filtering standards
std<- dplyr::filter(data_DF1, data_DF1$id=="STD")
std <- aggregate(std$blankminus ~ std$concentration, FUN = mean )
colnames (std) <-c("con", "OD")

## 3-parametric regression curve fitting
fit1<-nplr::nplr(std$con,std$OD,npars=3,useLog = FALSE)

## Linear regression curve fitting
fit2<- stats::lm(formula = con ~ OD,data = std)
```

```
## Estimating the 'blankminus'  
## eg:1 Based on nonparametric logistic regression fitting  
estimated_nplr <- estimate(data_DF1,colname = "blankminus",fitformula = fit1,method = "nplr")  
  
## eg:2 Based on linear regression fitting  
estimated_lr<-estimate(data_DF1,colname="blankminus",fitformula=fit2,method="linear")
```

extract_filename	<i>Extract Information From File Name</i>
------------------	---

Description

This function split a string (file name) as per the requirement of the user. It is useful to extract informations like compound name, plate number etc from the file name.

Usage

```
extract_filename(filename,split = " ",end = ".csv", remove = " ", sep = "-")
```

Arguments

filename	name of the file (string).
split	regular expressions at which filename has to be split to create different sections.
end	extension (end portion) of filename that need to be removed.
remove	section that need to be omitted after splitting the filename.
sep	symbol to be added to separate sections (obtained after splitting) before combining. default is "-"

Value

A character vector. First element is the unsplit 'filename'. Second element is the processed 'filename'. Other elements are different sections after splitting the 'filename'.

Author(s)

Anwar Azad Palakkan

Examples

```
extract_filename("L HEPG2 P3 72HRS.csv")  
extract_filename("L HEPG2 P3 72HRS.csv", split=" ",end=".csv",remove="L",sep="")
```

heatplate

Heatmap of multi well plate

Description

This function generate a heatmap (for numeric vector) or categorical plot (character vector) of multi well plate

Usage

```
heatplate(datamatrix, name, size = 7.5)
```

Arguments

datamatrix	data in matrix format. An easy way to create this is by calling matrix96
name	name to be given for the heatmap
size	plot size for each well in the heatmap (default is 7.5)

Details

Heat map can be generated for any multi well plate data in matrix format (datamatrix). The columns and rows of datamatrix should be labelled appropriately using [matrix96](#). A heatmap is generated if datamatrix is numeric, but a categorical plot is generated if datamatrix is a character matrix.

Value

A graphical plot.

Author(s)

A.A Palakkan

Examples

```
## loading data
data(metafile96, rawdata96,rawdata384)
rawdata96 <- data2plateformat(rawdata96,platetype = 96)
rawdata384 <- data2plateformat(rawdata384,platetype = 384)

## eg:1 heat map of rawdata96
data<-matrix96(plate2df(rawdata96),"value")
heatplate(data,"Plate 1", size=5)

## eg:2 heat map of rawdata96 can also be called as
heatplate(as.matrix(rawdata96),"Plate 1", size=5)

## eg:3 heat map of rawdata384
heatplate(as.matrix(rawdata384),"Plate 1", size=2)
```



```
## eg:4 catagorical map of metafile96 (column:id)
data<-matrix96(metafile96,"id")
heatplate(data,"Plate 1", size=5)
```

matrix96

Formatting Long Dataframe in to a Matrix Layout of Multi well Plate

Description

This function format a long dataframe (with col and row columns) in to a multiwell plate matrix layout.

Usage

```
matrix96 (dataframe, column, rm = "FALSE")
```

Arguments

dataframe	dataframe to be formatted
column	name of column (as a string in "") that need be converted as a matrix
rm	If rm = "TRUE" then -ve and NA are assigned as 0

Details

The 'dataframe' to be formatted should have a 'col' and 'row' columns representing the column and rowname of the corresponding multiwell plate.

Value

A matrix data with row and column names corresponding to multiwell plate

Author(s)

A.A Palakkan

Examples

```
## loading data
data(rawdata96, metafile96, metafile384)
rawdata<- data2plateformat(rawdata96, platetype = 96)
rawdata<- plate2df(rawdata)

## eg:1 rawdata to matrix format (column: value)
matrix96(rawdata,"value")

## eg:2 metafile96 to matrix format (column: id)
matrix96(metafile96,"id")
```

```
## eg:3 metafile384 to matrix format (column: cell)
matrix96(metafile384,"cell")
```

metafile384	<i>metadata of 384 well plate.</i>
-------------	------------------------------------

Description

A dataset containing metadata.

Usage

```
metafile384
```

Format

A data frame with 384 rows and 8 variables.

row Row number of multi well well plate.

col Column number of multi well plate.

position Well position address of multi well plate.

cell Type of cells used for the assay.

compound Different drugs (drug1,drug2,etc) used for the assay.

concentration 'C1','C2','C3' etc represent different concentration used for the same compound.
'B' represent blank wells

type 'treated' and 'untreated' shows if the wells had received pretreatment (example:inhibitors) or not. 'Blank1','Blank2','Blank3' etc represent separate blanks for different drugs.

dilution dilution of samples used for the assay.

Source

User generated metadata of the 384 well plate.

metafile96	<i>metadata of 96 well plate.</i>
------------	-----------------------------------

Description

A dataset containing metadata.

Usage

```
metafile96
```

Format

A data frame with 96 rows and 7 variables.

row Row number of multi well well plate.

col Column number of multi well plate.

position Well position address of multi well plate.

id Type of sample. 'STD' represent standards, 'sample' represent samples.

type Type of sample. 'STD1', 'STD2', 'STD3' etc represent different standards, 'S1', 'S2', 'S3' represent different samples.

concentration concentration of different standards (mg/ml).

dilution dilution of samples used for assay.

Source

User generated metadata of 96 well plate.

plate2df	<i>Format Matrix Type 2D Data of Multi well Plate as Dataframe</i>
----------	--

Description

This function uses column names and row names of 'datamatrix' (2D data of a mutli well plate) and generate a dataframe with row, col (column) and position indices. The 'value' column represent corresponding value in the 'datamarix'.

Usage

```
plate2df(datamatrix)
```

Arguments

datamatrix datamatrix is the 2D data of a mutli well plate. Usually the result of [data2plateformat](#):

Value

A dataframe with 4 columns. Number of rows is equal to the number of wells (plate type of 'datamatrix'). The columns represent

row	Row number of the entry
col	Column number of the entry
position	Position (Row+column number) of the entry
value	Individual entries in the 'datamatrix'

Author(s)

A.A Palakkan

Examples

```
## loading data
data(rawdata24,rawdata96,rawdata384)

## eg:1 spectrophotometer reading from 24 well plate in dataframe format
datamatrix<- data2plateformat(rawdata24, platetype = 24)
head(plate2df(datamatrix))

## eg:2 spectrophotometer reading from 96 well plate in dataframe format
datamatrix<- data2plateformat(rawdata96, platetype = 96)
head(plate2df(datamatrix))

## eg:3 spectrophotometer reading from 384 well plate in dataframe format
datamatrix<- data2plateformat(rawdata384, platetype = 384)
head(plate2df(datamatrix))
```

plate_metadata

Combining Plate Specific Information with Metadata

Description

plate_metadata combine the plate specific information (like compounds used, standard concentration, dilution of samples, etc) and metadata, to produce a plate specific metadata.

Usage

```
plate_metadata (plate_details, metadata, mergeby = "type")
```

Arguments

plate_details	plate specific information that need to be added to metadata
metadata	metadata for whole experiment
mergeby	column that is common to both metadata and plate_meta (as a string in "")

Details

plate_details need to be in a list format. Metadata should have a 'row' and 'col' columns representing the row and column names of the corresponding multi well plate.

Value

A dataframe. Each element of 'plate_details' will appear as a new column to the left of 'metadata'

Author(s)

A.A Palakkan

Examples

```
## loading data
data(metafile96)
plate_details <- list("compound" = "Taxol",
  "concentration" = c(0.00,0.01,0.02,0.05,0.10,1.00,5.00,10.00),
  "type" = c("S1","S2","S3","S4","S5","S6","S7","S8"),
  "dilution" = 1)

## eg:1 filling metadata96 using plate_details
plate_meta<-plate_metadata(plate_details,metafile96,mergeby="type")
head(plate_meta)
```

pvalue

t-Test on Summary Dataframe

Description

This function calculate the significance (t-test) within groups of 'dataframe'.

Usage

```
pvalue (dataframe, control, sigval)
```

Arguments

dataframe	a summary dataframe of dfsummary : output
control	control group name
sigval	pvalue cutoff for significance

Details

The 'dataframe' should be having similar format of `dfsummary` output. 'control' should be an element from the column just before 'label'. 'N', 'Mean', 'SD' and 'CV' columns in the 'dataframe' are used for calculating p value by t-test (one to one t-test with 'control' in that group). significant if pvalue is < 'sigval'. Different groups in 'dataframe' are evaluated separately (columns before label is used for grouping).

Value

A dataframe. New columns named 'pvalue' (p values of t-test.If the value is less than 0.001, then appear as "< 0.001") and 'significance' (yes if pvalue less than 'sigval') are attached to the left.

Author(s)

A.A Palakkan

Examples

```
## loading data
data(metafile384, rawdata384)
rawdata<-plate2df(data2plateformat(rawdata384,platetype = 384))
data_DF2<- dplyr::inner_join(rawdata,metafile384,by=c("row", "col", "position"))
result3 <- dfsummary(data_DF2,y = "value",
  grp_vector = c("cell", "compound", "concentration"),
  rm_vector = c("B", "drug2", "huh7"),
  nickname = "",
  rm = "FALSE", param = c(strict = "FALSE", cutoff = 40,n = 12))

## eg:1 t-test on result3.
pvalue(result3,"C3",sigval=0.05)
```

rawdata24

rawdata of 24 well plate.

Description

A dataset of spectrophotometer readings (OD).

Usage

rawdata24

Format

A data frame with 4 rows and 7 columns: First column shows 'row name' . .

Source

Spectrophotometer output reading (OD) of 24 well plate.

rawdata384	<i>rawdata of 384 well plate.</i>
------------	-----------------------------------

Description

A dataset of spectrophotometer readings (OD).

Usage

rawdata384

Format

A data frame with 16 rows and 25 columns: First column shows 'row name'.

Source

Spectrophotometer output reading (OD) of 384 well plate.

rawdata96	<i>rawdata of 96 well plate.</i>
-----------	----------------------------------

Description

A dataset of spectrophotometer readings (OD).

Usage

rawdata96

Format

A data frame with 8 rows and 13 columns: First column shows 'row name'.

Source

Spectrophotometer output reading (OD) of 96 well plate.

reduceblank	<i>Reduce Blank Values</i>
-------------	----------------------------

Description

This function can reduce 'blank' value from readings. Can handle separate blanks for separate groups in the dataframe

Usage

```
reduceblank(dataframe, x_vector, blank_vector, y)
```

Arguments

dataframe	Data in the form of dataframe.
x_vector	A character vector of groups/entries for which the blank has to be reduced.
blank_vector	A character vector of blank names whose value has to be reduced.
y	Name of the column (column should be numeric in nature) whose values has be reduced.

Details

This function will reduce the first blank vector element from first x_vector element and so on.

Value

A dataframe with a new column 'blankminus' (result of the blankminus function) added to the right.

Author(s)

A.A Palakkan

Examples

```
## loading data
data(metafile384, rawdata384)
rawdata<-plate2df(data2plateformat(rawdata384,platetype = 384))
data_DF2<- dplyr::inner_join(rawdata,metafile384,by=c("row","col","position"))

## eg:1 reduce blanks of data_DF2.
# reduce seperate blanks (mean of blank wells) for drug1, drug2, drug3 and drug4.
#blanks are blank1, blank2, blank3 and blank4 respectively for different drugs.

data_blk<-reduceblank(data_DF2,
  x_vector=c("drug1","drug2","drug3","drug4"),
  blank_vector = c("blank1","blank2","blank3","blank4"), "value")
```

rmodd_summary

Summarise a Numerical Vector with Control on Outlier Removal

Description

Summarise a numerical vector with control on how the outliers has to be treated.

Usage

```
rmodd_summary(x, rm = "FALSE", strict = "FALSE", cutoff = 80,n = 3)
```

Arguments

x	numerical vector
rm	if rm = "TRUE" outliers are omitted. If rm = "FALSE" all elements in the vector are considered for summarising
strict	if strict = "FALSE" outliers are omitted based on IQR rule. If strict = "TRUE" more aggressive outlier omitting method is used to bring CV below a cutoff value
cutoff	cv cutoff value for the aggressive outlier removal
n	minimum number of samples needed

Details

In IQR rule (ie when strict = "FALSE") those values above 'Q3 + 1.5 IQR' and those below 'Q1 - 1.5 IQR' is considered as outlier. For the aggressive outlier removal (ie when strict = "TRUE") those values above 90th percentile and below 10th percentile are removed consecutively till the cv fall below the 'cutoff' or only the minimum number of samples is leftover (whichever happens first halt the loop).

Value

A numeric vector of length 5 with the elements representing

mean	the average of samples
median	the median of samples
n	number of samples
sd	standard deviation of samples
cv	percentage cv of samples

Author(s)

A.A Palakkan

Examples

```
## data set x
x <- c(1.01,0.98,0.6,0.54,0.6,0.6,0.4,3)

## summarising without removing outliers
rmodd_summary(x, rm = "FALSE", strict= "FALSE", cutoff=80, n=3)

## summarising after removing outliers (IQR method)
rmodd_summary(x, rm = "TRUE", strict= "FALSE", cutoff=20, n=5)

## summarising after removing outliers (Stringent to reduce cv)
rmodd_summary(x, rm = "TRUE", strict= "TRUE", cutoff=20, n=5)
```

Index

- *Topic **arith**
 - dfsummary, 4
 - rmodd_summary, 17
- *Topic **character**
 - extract_filename, 7
- *Topic **datasets**
 - data_DF1, 4
 - metafile384, 10
 - metafile96, 11
 - rawdata24, 14
 - rawdata384, 15
 - rawdata96, 15
- *Topic **hplot**
 - heatplate, 8
- *Topic **htest**
 - pvalue, 13
- *Topic **manip**
 - data2plateformat, 3
 - matrix96, 9
 - plate2df, 11
 - plate_metadata, 12
 - reduceblank, 16
- *Topic **math**
 - estimate, 6

bioassays, 2

data2plateformat, 2, 3, 11

data_DF1, 4

dfsummary, 2, 4, 13, 14

estimate, 2, 6

extract_filename, 2, 7

heatplate, 2, 8

lm, 6

matrix96, 2, 8, 9

metafile384, 10

metafile96, 11

nplr, 6

plate2df, 2, 11

plate_metadata, 2, 12

pvalue, 2, 13

rawdata24, 14

rawdata384, 15

rawdata96, 15

reduceblank, 2, 16

rmodd_summary, 5, 17