

Package ‘gen5helper’

June 13, 2019

Type Package

Title Processing 'Gen5' 2.06 Exported Data

Version 1.0.1

Author Yanxian Lin [aut, cre]

Maintainer Yanxian Lin <yanxian00@gmail.com>

Description A collection of functions for processing 'Gen5' 2.06 exported data.

'Gen5' is a popular data analysis software for BioTek plate readers <<https://www.biotek.com/products/software-robotics-software/gen5-microplate-reader-and-imager-software/>>. This package contains functions for data cleaning, modeling and plotting using exported data from 'Gen5' version 2.06. It exports technically correct data defined in (Edwin de Jonge and Mark van der Loo (2013) <https://cran.r-project.org/doc/contrib/de_Jonge+van_der_Loo-Introduction_to_data_cleaning_with_R.pdf>) for customized analysis. It contains Boltzmann fitting for general kinetic analysis. It also implement line plot and bar plot for generating publishable figures. See <<https://www.github.com/yanxianUCSB/gen5helper>> for more information, documentation and examples.

Depends dplyr, utils

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports ggplot2, minpack.lm, plyr, pracma, tidyr, stats, natural sort,
rlang

NeedsCompilation no

R topics documented:

annotate	2
as.is	3
Boltzmann	3
Boltzmann_double	4
export2dataframe	4
factor2num	5
fit.boltzmann	5
g5h.annotate	6

g5h.annotate.deprecated	6
g5h.clean	7
g5h.clean2	7
g5h.clean_	8
g5h.gather_col	8
g5h.gather_row	9
g5h.map_col	9
g5h.map_row	10
g5h.set_time	10
g5h.set_time2	11
gather_col	11
gather_row	12
gen5helper	12
get.halftime	12
loadpkgs	13
map.group	13
mapvalues_	14
most.freq	14
normalize	15
range_	15
saveRDS_	16
smooth.mean	16
ui.clean	17
ui.fit.ThT	17
ui.plot.fit	17
ungroup_	18
write.csv_	18
Index	19

annotate	<i>Add useful variables</i>
----------	-----------------------------

Description

Add time interval, mean, standard deviation and initilized treatment and dose.

Usage

annotate(.data)

Arguments

.data data.frame cleaned by g5h.clean()

Value

data.frame appended with time intervals in minutes and hours, mean and standard deviation, grouped by col

as.is	<i>as.is</i>
-------	--------------

Description

as.is

Usage

as.is(x, vec)

Arguments

- | | |
|-----|-------------------------|
| x | object to transform |
| vec | object to extract class |

Boltzmann	<i>Boltzmann model for fitting time series data</i>
-----------	---

Description

Boltzmann model for fitting time series data

Usage

Boltzmann(time_, val_, A0 = 1, k0 = 1, t20 = 1)

Arguments

- | | |
|-------|------------------|
| time_ | time series |
| val_ | normalized value |
| A0 | amplititude |
| k0 | rate constant |
| t20 | halt time |

Value

a model

Boltzmann_double	<i>Boltzmann model for fitting time series data</i>
------------------	---

Description

Boltzmann model for fitting time series data

Usage

```
Boltzmann_double(time_, val_, start = list(y0 = 500, A = 500, k = 1.1, t2
= 10, A2 = 500, k2 = 1.1, t22 = 10))
```

Arguments

time_	NotExported
val_	NotExported
start	NotExported

Value

NotExported

export2dataframe	<i>Clean Gen5 exported data</i>
------------------	---------------------------------

Description

export2dataframe() returns technically correct data.frame from Gen5 2.06 exported tab-delim data. The exported data can be generated using default export protocol in Gen5 2.06. See Gen5 User Guide for more information.

Usage

```
export2dataframe(filename, Ctrl = list(sample.by = "row"))
```

Arguments

filename	the name of the file which the data are to be read from. If it does not contain an absolute path, the file name is relative to the current working directory, getwd().
Ctrl	list of controls. NOT IMPLEMENTED

Value

technically correct data.frame.

factor2num	<i>this is a file for functions that are universally useful at common data manipulations factor2num Convert factor to numeric</i>
------------	---

Description

this is a file for functions that are universally useful at common data manipulations factor2num
Convert factor to numeric

Usage

```
factor2num(x)
```

Arguments

x	factor
---	--------

Examples

```
factor2num(factor(c('1', '10', '100')))
```

fit.boltzmann	<i>Fit readings with Boltzmann model</i>
---------------	--

Description

fit.boltzmann() using Boltzmann model to fit readings and time intervals with unit of hours, using start as initial guesses. It appends A, y0, k, t2 and val.predict, while preserving existing variables.

Usage

```
fit.boltzmann(.data, A0 = 1, k0 = 1, t20 = 1)
```

Arguments

.data	data.frame with x as time, y as value
A0	initial guess of amplitue, default 1
k0	initial guess, default 1
t20	initial guess, default 1

Value

data.frame with fitted parameter and predicted value

g5h.annotate	<i>g5h.annotate Add time interval</i>
--------------	---------------------------------------

Description

g5h.annotate Add time interval

Usage

```
g5h.annotate(.data, by = "col")
```

Arguments

.data	data.frame cleaned by g5h.clean()
by	'col' or 'row', default is 'col'. See ?g5h.gather_col for more info.

Value

data.frame

g5h.annotate.deprecated	<i>Add useful variables</i>
-------------------------	-----------------------------

Description

Add time interval, mean, standard deviation and initilized treatment and dose.

Usage

```
g5h.annotate.deprecated(.data, by = "col")
```

Arguments

.data	data.frame cleaned by g5h.clean()
by	'col' or 'row', default is 'col'. See ?g5h.gather_col for more info.

Value

data.frame

g5h.clean	<i>Clean Gen5 exported data</i>
-----------	---------------------------------

Description

g5h.clean() returns technically correct data.frame from Gen5 2.06 exported tab-delim data. The exported data can be generated using default export protocol in Gen5 2.06. See Gen5 User Guide for more information.

Usage

```
g5h.clean(file)
```

Arguments

file	the name of the file which the data are to be read from. If it does not contain an absolute path, the file name is relative to the current working directory, getwd().
------	--

Value

technically correct data.frame.

g5h.clean2	<i>Clean Gen5 exported data</i>
------------	---------------------------------

Description

g5h.clean2() returns technically correct data.frame from Gen5 2.06 exported tab-delim data. The exported data can be generated using default export protocol in Gen5 2.06. See Gen5 User Guide for more information.

Usage

```
g5h.clean2(files)
```

Arguments

files	a vector of names of the file which the data are to be read from. If it does not contain an absolute path, the file name is relative to the current working directory, getwd().
-------	---

Value

technically correct data.frame.

g5h.clean_	<i>Clean Gen5 exported data</i>
------------	---------------------------------

Description

g5h.clean_() returns technically correct data.frame from Gen5 2.06 exported tab-delim data. The exported data can be generated using default export protocol in Gen5 2.06. See Gen5 User Guide for more information.

Usage

```
g5h.clean_(file)
```

Arguments

file	the name of the file which the data are to be read from. If it does not contain an absolute path, the file name is relative to the current working directory, getwd().
------	--

Value

technically correct data.frame.

g5h.gather_col	<i>Add mean and standard deviation</i>
----------------	--

Description

g5h.gather_col() preserve existing variables and add mean and standard deviation, grouped by col.
g5h.gather_row() preserve existing variables and add mean and standard deviation, grouped by row.

Usage

```
g5h.gather_col(.data)
```

Arguments

.data	data.frame
-------	------------

Value

data.frame appended with val.m and val.sd

g5h.gather_row	<i>Add mean and standard deviation</i>
----------------	--

Description

g5h.gather_col() preserve existing variables and add mean and standard deviation, grouped by col.
g5h.gather_row() preserve existing variables and add mean and standard deviation, grouped by row.

Usage

```
g5h.gather_row(.data)
```

Arguments

.data	data.frame
-------	------------

Value

data.frame appended with val.m and val.sd

g5h.map_col	<i>Map row or col to new values</i>
-------------	-------------------------------------

Description

g5h.map_row() or g5h.map_col() add new variables by mapping row or col to new values.

Usage

```
g5h.map_col(.data, feature, factors)
```

Arguments

.data	data.frame cleaned by g5h.clean()
feature	character, name of new variable to add.
factors	vector with length equal to levels of row or col.

Value

data.frame appended with new variables

g5h.map_row	<i>Map row or col to new values</i>
-------------	-------------------------------------

Description

g5h.map_row() or g5h.map_col() add new variables by mapping row or col to new values.

Usage

```
g5h.map_row(.data, feature, factors)
```

Arguments

.data	data.frame cleaned by g5h.clean()
feature	character, name of new variable to add.
factors	vector with length equal to levels of row or col.

Value

data.frame appended with new variables

g5h.set_time	<i>Add time intervals</i>
--------------	---------------------------

Description

g5h.set_time() preserves existing variables and add new, realMinute and realHour, which are time intervals in minutes and hours.

Usage

```
g5h.set_time(.data)
```

Arguments

.data	data.frame cleaned by g5h.clean()
-------	-----------------------------------

Value

input data.frame appended with realMinute and realHour

g5h.set_time2	<i>Add time intervals</i>
---------------	---------------------------

Description

g5h.set_time() preserves existing variables and add new variable, time, which are time intervals in hours.

Usage

```
g5h.set_time2(.data, units = "hours")
```

Arguments

.data	data.frame cleaned by g5h.clean()
units	hours or minutes

Value

input data.frame appended with time

gather_col	<i>Add mean and standard deviation</i>
------------	--

Description

gather_col() preserve existing variables and add mean and standard deviation, grouped by col.
gather_row() preserve existing variables and add mean and standard deviation, grouped by row.

Usage

```
gather_col(.data)
```

Arguments

.data	data.frame
-------	------------

Value

data.frame appended with val.m and val.sd

gather_row	<i>Add mean and standard deviation</i>
------------	--

Description

gather_col() preserve existing variables and add mean and standard deviation, grouped by col.
gather_row() preserve existing variables and add mean and standard deviation, grouped by row.

Usage

```
gather_row(.data)
```

Arguments

.data	data.frame
-------	------------

Value

data.frame appended with val.m and val.sd

gen5helper	<i>gen5helper: A Collection of Functions for Processing Gen5 2.06 Exported Data</i>
------------	---

Description

A collection of functions for processing Gen5 2.06 exported data. Gen 5 is a popular data analysis software for BioTek plate readers. This packages contains functions for data cleaning, modeling and plotting using exported data from Gen5 version 2.06. It exports technically correct data defined in (Edwin de Jonge and Mark van der Loo, 2013) for customized analysis. It contains Boltzmann fitting for general kinetic analysis. It also implement line plot and bar plot for generating publishable figures. See <https://www.github.com/yanxianUCSB/gen5helper> for more information, documentation and examples.

get.halftime	<i>Get half time by linear fitting</i>
--------------	--

Description

get half time according to <http://www.amylofit.ch.cam.ac.uk> The algorithm for the extraction of the half times proceeds as follows: first the middle part of the curve is selected, by determining when the average over several points is first above 0.3 and when the average is last below 0.7. The number of points to be averaged over depends on the number of points in the curve. A straight line is then fitted to this middle part of the curve, the point at which it crosses the value of 0.5 is recorded as the half time. (source: DOI: nprot.2016.010)

Usage

```
get.halftime(time, val)
```

Arguments

time	vector of time
val	vector of values

Value

half time

loadpkgs	<i>Attach common packages</i>
----------	-------------------------------

Description

Attach common packages

Usage

```
loadpkgs()
```

map.group	<i>Map group into new variable</i>
-----------	------------------------------------

Description

Map group into new variable

Usage

```
map.group(.data, newvar, facs)
```

Arguments

.data	data.frame with group info
newvar	name of new variable to add.
facs	vector with length equal to levels of group

Value

data.frame with added new variable

mapvalues_	<i>mapvalues_</i>
------------	-------------------

Description

mapvalues_

Usage

```
mapvalues_(x, facs, bNaturalSort = F)
```

Arguments

x	factor or character
facs	character. It maps unique(x) to facs
bNaturalSort	binary

Value

factor

most.freq	<i>Most frequent numbers get the n most frequent elements in an array</i>
-----------	---

Description

Most frequent numbers get the n most frequent elements in an array

Usage

```
most.freq(x, n = 1)
```

Arguments

x	an array of elements
n	integer, default is 1

Value

the most n elements

Examples

```
most.freq(c('a', 'a', 'b', 'b', 'b', 'c'), n = 2)
most.freq(c(1, 1, 2, 3, 3, 3, 4, 4), n = 2)
```

normalize	<i>Normalize a vector by min and max</i>
-----------	--

Description

NA removed

Usage

```
normalize(x)
```

Arguments

x	numeric
---	---------

Value

a normalized vector

Examples

```
normalize(0:10)
```

range_	<i>Range of a vector</i>
--------	--------------------------

Description

This computes the range of a vector as a value; NA removed.

Usage

```
range_(x, na.rm = T)
```

Arguments

x	numeric
na.rm	bool whether to remove NA values.

Value

numeric value

Examples

```
range_(c(1, 5, 10))
```

saveRDS_	<i>saveRDS and return .data</i>
----------	---------------------------------

Description

saveRDS and return .data

Usage

```
saveRDS_(.data, file, ...)
```

Arguments

.data	object to be saved
file	filename to save
...	for saveRDS

Value

.data

smooth.mean	<i>Smooth a vector using moving average</i>
-------------	---

Description

Smooth a vector using moving average

Usage

```
smooth.mean(vec, naverage)
```

Arguments

vec	numeric vector
naverage	width of moving average

Value

smoothed

Examples

```
smooth.mean(1:10, 2)
smooth.mean(1:10, 3)
smooth.mean(1:10, 5)
```

ui.clean	<i>UI for g5h.clean</i>
----------	-------------------------

Description

ui.clean() is a command line user interface for g5h.clean. See ?g5h.clean for more information.

Usage

```
ui.clean(args = commandArgs(trailingOnly = T))
```

Arguments

args	command arguments
------	-------------------

Value

data.frame

ui.fit.ThT	<i>UI fit ThT</i>
------------	-------------------

Description

User interface for fitting ThT kinetic curve

Usage

```
ui.fit.ThT(START = list(A = 3000, y0 = 1000, k = 1, t2 = 5))
```

Arguments

START	initial guess for fit.Boltzmann
-------	---------------------------------

ui.plot.fit	<i>UI plot fit</i>
-------------	--------------------

Description

User interface for plotting fitted ThT kinetic

Usage

```
ui.plot.fit()
```

ungroup_	<i>Ungroup() and as.data.frame()</i>
----------	--------------------------------------

Description

Ungroup() and as.data.frame()

Usage

```
ungroup_(.data)
```

Arguments

.data	grouped data.frame
-------	--------------------

Value

data.frame()

write.csv_	<i>write.csv and return .data</i>
------------	-----------------------------------

Description

write.csv and return .data

Usage

```
write.csv_(x, file)
```

Arguments

x	object
file	filename for write.csv

Value

x

Index

annotate, [2](#)
as.is, [3](#)

Boltzmann, [3](#)
Boltzmann_double, [4](#)

export2dataframe, [4](#)

factor2num, [5](#)
fit.boltzmann, [5](#)

g5h.annotate, [6](#)
g5h.annotate.deprecated, [6](#)
g5h.clean, [7](#)
g5h.clean2, [7](#)
g5h.clean_, [8](#)
g5h.gather_col, [8](#)
g5h.gather_row, [9](#)
g5h.map_col, [9](#)
g5h.map_row, [10](#)
g5h.set_time, [10](#)
g5h.set_time2, [11](#)
gather_col, [11](#)
gather_row, [12](#)
gen5helper, [12](#)
gen5helper-package (gen5helper), [12](#)
get.halftime, [12](#)

loadpkgs, [13](#)

map.group, [13](#)
mapvalues_, [14](#)
most.freq, [14](#)

normalize, [15](#)

range_, [15](#)

saveRDS_, [16](#)
smooth.mean, [16](#)

ui.clean, [17](#)
ui.fit.ThT, [17](#)
ui.plot.fit, [17](#)
ungroup_, [18](#)

write.csv_, [18](#)