Package 'titer'

December 1, 2016

Title Tools for analyzing and visualizing antibody titer data

Version 0.0.2.0009

Description This package contains methods to calculate endpoints from antibody titer data and visualize titers.

Depends R (>= 3.0.2)

Imports dplyr,
ggplot2,
grid,
tidyr

BugReports https://github.com/stefanavey/titer/issues

License MIT + file LICENSE

LazyData true

RoxygenNote 5.0.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`+.uneval` *Addition for* `aes()` *and* `aes_string()`

Description

`+.uneval` is a helper function to allow adding aes and aes_string in ggplot2

Usage

```
## S3 method for class 'uneval'
a + b
```

Arguments

`a`	first argument
`b`	second argument

References

http://stackoverflow.com/questions/28777626/how-do-i-combine-aes-and-aes-string-options

---

`Barplot` *Titer bar plots.*

Description

`Barplot` plots the baseline and day 28 titers

Usage

```
Barplot(dat_list, subjectCol = "SubjectID", cols = 1, groupVar = NULL,
  colors = c("#A6CEE3", "#1F78B4", "#B2DF8A", "#33A02C", "#FB9A99", "#E31A1C",
  "#FDBF6F", "#FF7F00"))
```

Arguments

`dat_list`	a named list like the one returned by `FormatTiters`.
`subjectCol`	the name of the column specifying a subject ID. Default is "SubjectID".
`cols`	numeric specifying how many columns to layout plot
`groupVar`	an optional character string specifying a grouping variable. May be either a variable in `dat_list` or an endpoint. Default is `NULL`
`colors`	a vector of colors specifying bar colors. If `dat_list` contains more than 4 elements, you must specify your own colors.

Value

(invisibly) a list of ggplot2 object(s).

Author(s)

Stefan Avey

Examples

```
## Prepare the data
titer_list <- FormatTiters(Year1_Titers)

## Bar plot of a single strain
Barplot(titer_list["A California 7 2009"])

## Bar plot of all 3 strains
Barplot(titer_list)

## Can improve readability of previous plot by separating into groups
## For example, group by AgeGroup
Barplot(titer_list, groupVar = "AgeGroup")
```

`BubbleChart` *Bubble Chart*

Description

`BubbleChart` visualizes baseline vs fold change in titers

Usage

```
BubbleChart(dat_list, subjectCol = "SubjectID", fit = NULL,
  yMinZero = FALSE, eqSize = 6/log2(length(dat_list) + 1), colorBy = NULL,
  xlimits = c(1.5, 10.5), xbreaks = 2:10, ylimits = c(-0.5, 10),
  ybreaks = seq(0, 10, 2), plot = TRUE, cols = 2, ...)
```

Arguments

`dat_list`	a named list like the one returned by `FormatTiters`. Values are assumed to be log2-transformed.
`subjectCol`	the name of the column specifying a subject ID. Default is "SubjectID".
`fit`	what type of fit to add. Current options are "lm" for linear model, "exp" for exponential, or `NULL` for no smoothing.
`yMinZero`	a logical specifying whether fitted y values below 0 should be set to 0.
`eqSize`	Text size of the equation. Only relevant if `fit` is not `NULL`
`colorBy`	a character string specifying an endpoint to colorBy or `NULL` (default) for no coloring.
`xlimits`	the x-axis limits (passed to `scale_x_continuous`)
`xbreaks`	the x-axis breaks (passed to `scale_x_continuous`)
`ylimits`	the y-axis limits (passed to `scale_y_continuous`)
`ybreaks`	the y-axis breaks (passed to `scale_y_continuous`)
`plot`	logical indicating whether to plot or not. Default is TRUE
`cols`	numeric specifying how many columns to layout plot
`...`	other arguments besides `method` and `subjectCol` passed to `CalculatemaxRBA`.

Details

This plot was designed for HAI titer data with baseline columns and fold change columns for multiple strains.

Value

(invisibly) a list of ggplot2 objects.

Author(s)

Stefan Avey

See Also

`FormatTiters`

Examples

```
## Prepare the data
titer_list <- FormatTiters(Year2_Titers)

## Basic plot without any fitted model
BubbleChart(titer_list)

## Change layout to plot all in a single column
BubbleChart(titer_list, cols = 1)

## Add a linear fit
BubbleChart(titer_list, fit = "lm")

## Add an exponential fit
BubbleChart(titer_list, fit = "exp")

## Add coloring by age
BubbleChart(titer_list, fit = "exp", colorBy = "AgeGroup")
```

`CalculateD0NormPaired` *CalculateD0NormPaired*

Description

`CalculateD0NormPaired` calculates the normalized day 0 titer paired with the titer with maximum normalized fold change

Usage

```
CalculateD0NormPaired(dat, fcStdCols = grep("fc_std_norm", colnames(dat),
  value = TRUE))
```

Arguments

`dat`	data frame containing `fcStdCols`
`fcStdCols`	column names containing the titer fold changes for each strain standardized across subjects

Details

If there are multiple strains that have the maximal fold change, choose the day 0 titer that is higher since this will allow for a greater adjustment and better chance of being a high responder.

Column names containing the day 0 titers for each strain standardized across subjects are assumed to follow the same pattern as `fcStdCols` with "d0" replacing "fc" in the name.

Value

a numeric vector containing the values from d0StdCols that correspond to the maximum over the strains of fcStdCols

Author(s)

Stefan Avey

Examples

```
## First Example
```

`CalculatemaxRBA` *Calculate maxRBA*

Description

`CalculatemaxRBA` calculates the maximum residual after baseline-adjustment for each viral strain

Usage

```
CalculatemaxRBA(dat_list, subjectCol = "SubjectID", method = c("exp", "lm"),
  yMinZero = FALSE, scoreFun = max, discretize = c(0.2, 0.3),
  normalize = FALSE, scaleResiduals = FALSE,
  responseLabels = paste0(c("low", "moderate", "high"), "Responder"),
  na_action = "na.fail", ...)
```

Arguments

`dat_list`	a named list like the one returned by `FormatTiters`.
`subjectCol`	the name of the column specifying a subject ID. Default is "SubjectID".
`method`	a character string specifying the method used to model the relationship between day 0 and fold change values. One of either "lm" for a linear model or "exp" for an exponential model.
`yMinZero`	a logical specifying whether fitted y values below 0 should be set to 0.
`scoreFun`	a function applied to all (potentially scaled) residuals for each subject to determine the endpoint. Default is `max` but `sum` may also be useful to quantify the total response.
`discretize`	a vector of quantiles in (0, 0.5] specifying where to make the cutoff for low, moderate and high responses. Default is 20% and 30%.
`normalize`	Logical specifying whether residuals should be normalized with the inverse normal transform. Default is `FALSE`.

scaleResiduals	Logical. Should residuals be scaled inversely by the square of the confidence intervals from the linear model.
responseLabels	names for low, moderate and high responses
na_action	how should missing NA values be treated. Default is "na.fail"
...	Additional arguments passed to lm if method == "lm" or nls if method == "exp"

Details

Calculates the baseline-adjusted fold change for each strain of virus using (unnormalized) fold change and baseline titers. Linear regression or an exponential curve is used to remove the effect of baseline titers on fold changes. The score function (scoreFun) is used to combine the adjusted fold change across multiple strains. Missing (NA) values are handled by being returned as missing in the endpoints in the output

Value

A list with the following elements:

models the models calculated on each strain separately (with names the same as on dat_list)

residualMatrix the matrix of residuals

maxRBA_d<X> a named vector containing the discrete maxRBA endpoint with a cutoff at <X>

... Other named vectors containing discrete maxRBA endpoints

Author(s)

Stefan Avey

See Also

lm, nls

Examples

```
## Prepare the data
titer_list <- FormatTiters(Year2_Titers)

## Using a linear fit
endpoints <- CalculatemaxRBA(titer_list, method = "lm")
summary(endpoints)
## Get discrete endpoints using upper/lower 30%
endpoints$maxRBA_d30

## Get endpoints with a 50% split into high and low
endpoints <- CalculatemaxRBA(titer_list, method = "exp", discretize = 0.5)
endpoints$maxRBA_d50
```

`CalculateMFC` *Calculate MFC*

Description

`CalculateMFC` calculates the (log-transformed) maximum fold change over all strains.

Usage

```
CalculateMFC(dat_list, subjectCol = "SubjectID", discretize = c(0.2, 0.3),
  responseLabels = paste0(c("low", "moderate", "high"), "Responder"))
```

Arguments

`dat_list`	a named list like the one returned by `FormatTiters`.
`subjectCol`	the name of the column specifying a subject ID. Default is "SubjectID".
`discretize`	a vector of quantiles in (0, 0.5] specifying where to make the cutoff for low, moderate and high responses. Default is 20% and 30%.
`responseLabels`	names for low, moderate and high responses

Value

A list with the following elements:

MFC a named vector containing the continuous MFC endpoints

MFC_d<X> a named vector containing the discrete MFC endpoint with a cutoff at <X>

... Other named vectors containing discrete MFC endpoints

A named vector containing the MFC for each subject

Author(s)

Stefan Avey

Examples

```
## Prepare the data
titer_list <- FormatTiters(Year2_Titers)

CalculateMFC(titer_list)
```

CalculatePadjMFC *CalculatePadjMFC*

Description

CalculatePadjMFC calculates the paired, adjusted maximum fold change (padjMFC)

Usage

```
CalculatePadjMFC(dat, fcCol = "fc_norm_max_ivt", d0Col = "d0_norm_paired",
  discretize = c(0.2, 0.3), scaleResiduals = FALSE,
  responseLabels = paste0(c("low", "moderate", "high"), "Responder"), ...)
```

Arguments

dat	the data containing the columns fcCol and d0Col
fcCol	character string specifying the name of the fold change column from dat
d0Col	character string specifying the name of the day 0 column from dat
discretize	a vector of quantiles in (0, 0.5] specifying where to make the cutoff for low, moderate and high responses. Default is 20% and 30%.
scaleResiduals	Logical. Should residuals be scaled inversely by the square of the confidence intervals from the linear model.
responseLabels	names for low, moderate and high responses
...	Additional arguments passed to lm

Details

Calculate the paired, adjusted maximum fold change (padjMFC) from fc_norm_max_ivt and d0_norm_paired using linear regression to remove the effect of baseline titers. Missing (NA) values are handled and any missing values in fcCol and d0Col will also be missing in the output.

Value

A list with the first element named "linearModel" for the linear model and then "padjMFC" containing the continuous padjMFC metric and one additional element for each value of discretize giving the discrete labels.

Author(s)

Stefan Avey

See Also

lm

Examples

```
## First Example
```

CalculatepreGMT *Calculate pre-GMT*

Description

CalculatepreGMT calculates the log-transformed pre-vaccination geometric mean titer (pre-GMT)

Usage

```
CalculatepreGMT(dat_list, subjectCol = "SubjectID")
```

Arguments

dat_list	a named list like the one returned by FormatTiters.
subjectCol	the name of the column specifying a subject ID. Default is "SubjectID".

Details

Non-logged HAI titers for each strain are used to calculate the geometric mean and the geometric mean for each subject is subsequently log2-transformed.

Value

A named vector containing the pre-GMT for each subject

Author(s)

Stefan Avey

Examples

```
## Prepare the data
titer_list <- FormatTiters(Year2_Titers)

CalculatepreGMT(titer_list)
```

CalculateStdNorm *Calculate Normalized Titers*

Description

CalculateStdNorm calculates the standardized d0 or fc titers

Usage

```
CalculateStdNorm(dat, type, fcToOne = FALSE, idCol = "SubjectID",
  cols = grep(paste0(type, "_[AB]"), colnames(dat), value = TRUE))
```

Arguments

dat	Data frame containing fcStdCols
type	What should be standarized. Either "d0", or "fc".
fcToOne	Logical. Are titer fold changes allowed to be less than 1 or should these be changed to 1 before standardization? Default is FALSE and no changes will be made. Only relevant when type == "fc"
idCol	Name of column containing subject IDs
cols	column names containing the titer measurements for each strain

Details

This must be run on only 1 cohort at a time because titers will be normalized across all subjects. The median is used but unlike the original reference, the standard deviation is calculated rather than the maximum absolute deviation.

Value

A data frame like dat but with standarized columns added

Author(s)

Stefan Avey

References

Tsang JS, et al. (2014) Global analyses of human immune variation reveal baseline predictors of postvaccination responses. Cell 157(2):499-513.

Examples

```
## First Example
```

CalculatewhoResp *Calculate whoResp*

Description

CalculatewhoResp calculates a response definition similar to the WHO defintion using a 4-fold cutoff.

Usage

```
CalculatewhoResp(dat_list, subjectCol = "SubjectID")
```

Arguments

dat_list	a named list like the one returned by FormatTiters.
subjectCol	the name of the column specifying a subject ID. Default is "SubjectID".

Details

Subjects are responders ("R") if they acheive a 4-fold or greater fold change in titer to at least 2 strains, nonresponders ("NR") if they do not acheive a 4-fold or greater fold change in titer to any strain, and intermediate ("X") otherwise. Missing (NA) values are handled by being returned as missing in the endpoints in the output

Value

A named vector containing the response ("NR", "X", or "R") for each subject

Author(s)

Stefan Avey

Examples

```
## Prepare the data
titer_list <- FormatTiters(Year2_Titers)

CalculatewhoResp(titer_list)
```

FormatTiters *Format antibody titers.*

Description

FormatTiters formats titers into a list with one tidy data frame per viral strain

Usage

```
FormatTiters(titers, log2Transform = TRUE, fcMinZero = TRUE)
```

Arguments

titers	a data frame containing one row per subject per strain. The following columns are required: **SubjectID** Subject IDs (column name can vary) **Strain** The name of the viral strain for the observation **Pre** The pre-vaccination (or pre-infection) titer **Post** The post-vaccination (or post-infection) titer **...** Other columns which will be preserved
log2Transform	logical specifying whether titer values should be log2 transformed
fcMinZero	should negative fold changes be set to 0? Default is TRUE

Value

a list of data frames with one data frame per viral strain containing the "Pre" and "Post" titer measurements (row names are removed).

Author(s)

Stefan Avey

Examples

```
titer_list <- FormatTiters(Year1_Titers, log2Transform = TRUE, fcMinZero = TRUE)
```

FormatTiters_OLD *Format antibody titers.*

Description

`FormatTiters` formats titers into a list with one tidy data frame per viral strain

Usage

```
FormatTiters_OLD(titers, strains, subjectCol = "SubjectID",
  otherCols = vector(mode = "character"), d0Cols = paste0("d0_", strains),
  fcCols = paste0("fc_", strains), fcMinZero = TRUE, log2Transform = TRUE)
```

Arguments

`titers`	a data frame containing the titer information
`strains`	the names of the virus strains
`subjectCol`	the name of the column specifying a subject ID. Default is "SubjectID".
`otherCols`	a character vector specifying which additional columns of titers to retain. (Defaults to an empty character vector).
`d0Cols`	the column names of day 0 (baseline) columns
`fcCols`	the column names of fold change columns
`fcMinZero`	should negative fold changes be set to 0? Default is `TRUE`
`log2Transform`	logical specifying whether titer values should be log2 transformed

Value

a list of data frames with one data frame per viral strain containing the baseline ("d0"), fold change ("fc") and any other columns specified by the `otherColumns` argument.

Author(s)

Stefan Avey

Examples

```
strains <- c("A_California_7_2009", "A_Perth_16_2009", "B_Brisbane_60_2008")
titer_list <- FormatTiters(Year1_Titers, strains, subjectCol = "YaleID")
```

GetEqn *Get Formatted Model Equation*

Description

GetEqn gets the equation for various models in a human readable format

Usage

```
GetEqn(m)
```

Arguments

m	a model object

Author(s)

Stefan Avey

References

original lm_eqn and inspiration from this SO post http://stackoverflow.com/questions/7549694/ggplot2-adding-regression-line-equation-and-r2-on-graph.

Multiplot *Multiple ggplot2 plots on the same page*

Description

Multiple Plot Function for ggplot

Usage

```
Multiplot(..., plotlist = NULL, cols = 1, layout = NULL)
```

Arguments

...	ggplot objects
plotlist	a list of ggplot objects
cols	Number of columns in layout
layout	A matrix specifying the layout. If present, 'cols' is ignored

Details

If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE), then plot 1 will go in the upper left, 2 will go in the upper right, and 3 will go all the way across the bottom.

Author(s)

R Cookbook

References

http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_%28ggplot2%29/

`titeR` *titeR - An R package for antibody titer data*

Description

titeR - An R package for antibody titer data

`Year1_Titers` *Year 1 titers.*

Description

Antibody titers to 3 strains of influenza in a cohort of young and older adults from Yale during the 2010-2011 flu season.

Usage

```
Year1_Titers
```

Format

A data frame with 42 rows and 11 variables:

SubjectID a unique subject identifier

AgeGroup age of subject. 20-35 (Young), 65+ (Older)

Strain The name of the viral strain for the observation

Pre The pre-vaccination titer

Post The post-vaccination titer

References

Thakar J, et al. (2015) Aging-dependent alterations in gene expression and a mitochondrial signature of responsiveness to human influenza vaccination. Aging (Albany NY) 7(1):38-52. https://www.ncbi.nlm.nih.gov/pubmed/25596819

Year2_Titers *Year 2 titers.*

Description

Antibody titers to 3 strains of influenza in a cohort of young and older adults from Yale during the 2011-2012 flu season.

Usage

```
Year2_Titers
```

Format

A data frame with 69 rows and 11 variables:

SubjectID a unique subject identifier

AgeGroup age of subject. 20-35 (Young), 65+ (Older)

Strain The name of the viral strain for the observation

Pre The pre-vaccination titer

Post The post-vaccination titer

References

Thakar J, et al. (2015) Aging-dependent alterations in gene expression and a mitochondrial signature of responsiveness to human influenza vaccination. Aging (Albany NY) 7(1):38-52. https://www.ncbi.nlm.nih.gov/pubmed/25596819

Index