Package 'titeR'

November 21, 2016

Title Tools for analyzing and visualizing antibody titer data.

Version 0.0.1.0004

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

Depends R (>= 3.0.2)

License CCO

LazyData true

RoxygenNote 5.0.1

R topics documented:

`BubbleChart` *Bubble Chart*

Description

`BubbleChart` visualizes baseline vs fold change in titers

Usage

```
BubbleChart(dat_list, fit = NULL, xlimits = c(1.5, 10.5), xbreaks = 2:10,
  plot = TRUE, cols = 2)
```

Arguments

dat_list	a list like the one returned by FormatTiters
fit	what type of fit to add. Current options are "lm" for linear model, "exp" for exponential, or NULL for no smoothing.
xlimits	the x-axis limits (passed to scale_x_continuous)
xbreaks	the x-axis breaks (passed to scale_x_continuous)
plot	logical indicating whether to plot or not. Default is TRUE
cols	numeric specifying how many columns to layout plot
scale_y	a character string specifying whether the y axis should be "fixed" for all strains or "free".

Details

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

Value

a list of ggplot2 objects.

Author(s)

Stefan Avey

See Also

FormatTiters

Examples

```
## Not run:
## Example using the master phenotype file
library(dplyr)
titers <- master %>%
  filter(Year == 1, AgeGroup %in% "Young", !is.na(whoResp))
titer_list <- FormatTiters(titers,
                           strains = c("A_California_7_2009",
                               "A_Perth_16_2009",
                               "B_Brisbane_60_2008"))

## End(Not run)
BubbleChart(titer_list)
```

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
```

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
```

`CalculateSAdjMFC` *Calculate SAdjMFC*

Description

`CalculateSAdjMFC` calculates the baseline-adjusted maximum fold change (MFC) for each viral strain

Usage

```
CalculateSAdjMFC(datList, subjectCol = "SubjectID", method = c("lm", "exp"),
  scoreFun = max, fcCol = "fc", d0Col = "d0", normalize = TRUE,
  discretize = c(0.2, 0.3), scaleResiduals = FALSE,
  responseLabels = paste0(c("low", "moderate", "high"), "Responder"),
  na_action = "na.fail", ...)
```

Arguments

`datList`	a list with one data frame for each strain and each data frame containing the columns `fcCol` and `d0Col`. The order of each data frame must be the same and they must be the same dimensions. In addition, each data frame must be sorted by `d0Col` from low to high.
`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.
`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.
`fcCol`	character string specifying the name of the fold change column in each element of `datList`
`d0Col`	character string specifying the name of the day 0 column in each element of `datList`
`normalize`	Logical specifying whether residuals should be normalized with the inverse normal transform. Default is `TRUE`.
`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
`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 `datList`) "residualMatrix": the matrix of residuals "SAdjMFC": a list containing the continuous and discrete SAdjMFC metrics

Author(s)

Stefan Avey

See Also

`lm, nls`

Examples

```
## First Example
```

`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<e2><80><93>513.

Examples

```
## First Example
```

FormatTiters *Format antibody titers.*

Description

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

Usage

```
FormatTiters(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

```
## Not run:
## Example using the master phenotype file
library(dplyr)
titers <- master %>%
  filter(Year == 1, AgeGroup %in% "Young", !is.na(whoResp))
titer_list <- FormatTiters(titers,
                           strains = c("A_California_7_2009",
                               "A_Perth_16_2009",
                               "B_Brisbane_60_2008"))

## End(Not run)
```

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.

Examples

```
## First Example
```

`Multiplot` *Multiple ggplot2 plots on the same page*

Description

Multiple Plot Function for ggplot

Usage

```
Multiplot(..., plotlist = NULL, file, 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/

Examples

```
library(ggplot2)

## This example uses the ChickWeight dataset, which comes with ggplot2
## First plot
p1 <- ggplot(ChickWeight, aes(x=Time, y=weight, colour=Diet, group=Chick)) +
  geom_line() +
  ggtitle("Growth curve for individual chicks")

                                       # Second plot
p2 <- ggplot(ChickWeight, aes(x=Time, y=weight, colour=Diet)) +
  geom_point(alpha=.3) +
  geom_smooth(alpha=.2, size=1) +
  ggtitle("Fitted growth curve per diet")

                                       # Third plot
p3 <- ggplot(subset(ChickWeight, Time==21), aes(x=weight, colour=Diet)) +
  geom_density() +
  ggtitle("Final weight, by diet")
```

```
                                        # Fourth plot
p4 <- ggplot(subset(ChickWeight, Time==21), aes(x=weight, fill=Diet)) +
  geom_histogram(colour="black", binwidth=50) +
  facet_grid(Diet ~ .) +
  ggtitle("Final weight, by diet") +
  theme(legend.position="none")        # No legend (redundant in this graph)

Multiplot(p1, p2, p3, p4, cols=2)
```

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

Description

titeR - An R package for antibody titer data

Index