

Package ‘Glimma’

October 12, 2016

Type Package

Title Interactive HTML graphics for RNA-seq data

Version 1.0.0

Date 2016-02-29

Author Shian Su, Matthew E. Ritchie

Maintainer Shian Su <su.s@wehi.edu.au>

Description This package generates interactive visualisations of RNA-sequencing data based on output from limma, edgeR or DESeq2. Interactions are built on top of popular static displays from the limma package, providing users with access to gene IDs and sample information. Plots are generated using d3.js and displayed in HTML pages.

biocViews DifferentialExpression, ReportWriting, RNASeq, Visualization

Depends R (>= 3.2.0)

Imports DESeq2, edgeR, grDevices, methods, stats, utils

Suggests BiocStyle, limma

License GPL-3 | file LICENSE

URL <https://github.com/Shians/Glimma>

BugReports <https://github.com/Shians/Glimma/issues>

RoxygenNote 5.0.1

NeedsCompilation no

LazyData true

R topics documented:

CharToHexCol	2
glAutoinput	3
glBar	3
glBar.default	4
glimma	5
glink	6
glMDPlot	7

glMDPlot.default	8
glMDPlot.DESeqDataSet	9
glMDPlot.DESeqResults	10
glMDPlot.DGEEExact	12
glMDPlot.DGELRT	13
glMDPlot.MArrayLM	14
glMDSPlot	16
glMDSPlot.default	17
glMDSPlot.DGEList	18
glScatter	19
glScatter.default	19
hasColumns	21
lymphomaRNAseq	21
makeChartJson	22
makeDFJson	22
NumToHexCol	23
Index	24

CharToHexCol	<i>String to hex colour converter</i>
--------------	---------------------------------------

Description

Function to convert colour strings into hex codes

Usage

```
CharToHexCol(x)
```

Arguments

x the string colour value(s) to be converted to hex values.

Value

hex codes for colours

glAutoinput	<i>Glimma Autocomplete Input</i>
-------------	----------------------------------

Description

Create an autocompleted input field.

Usage

```
glAutoinput(target, action, idval = "none", flag = "none")
```

Arguments

target	the index of the plot which receives the event and performs an action.
action	the action to be performed at target plot using input information.
idval	the column from which the autocomplete list will be populated.
flag	indicates special flags for custom features.

Value

a input object containing the input field information.

glBar	<i>Glimma MD Plot</i>
-------	-----------------------

Description

Create an interactive bar plot object.

Usage

```
glBar(x, ...)
```

Arguments

x	the data.frame containing data to plot.
...	additional arguments depending on input object type.

Value

A chart object containing the information to create an interactive bar plot.

Author(s)

Shian Su

See Also[glBar.default](#)**Examples**

```

data(mtcars)
counts <- table(mtcars$gear)
data <- data.frame(nGears=as.numeric(names(counts)), Count=as.numeric(counts))

plot1 <- glBar(data, "Count", "nGears", ylab="Number of Gears")
glimma(plot1, layout=c(1,1), launch=TRUE)

```

`glBar.default`*Glimma MD Plot*

Description

Default method for interactive bar plot.

Usage

```

## Default S3 method:
glBar(x, yval, names.arg = rownames(x), ndigits = NULL,
      signif = 6, xlab = NULL, ylab = yval, main = NULL, height = 400,
      width = 500, colval = NULL, annot = yval, flag = NULL, info = NULL,
      ...)

```

Arguments

<code>x</code>	the data.frame containing data to plot.
<code>yval</code>	the column name for the x-axis values.
<code>names.arg</code>	the column name for the label on each bar.
<code>ndigits</code>	the number of digits after the decimal to round to in the tooltip (overrides <code>signif</code>).
<code>signif</code>	the number of significant figures to display in the tooltip.
<code>xlab</code>	the label on the x-axis.
<code>ylab</code>	the label on the y-axis.
<code>main</code>	the title for the plot.
<code>height</code>	the height of the plot (in pixels).
<code>width</code>	the width of the plot (in pixels).
<code>colval</code>	the colours for each data point.
<code>annot</code>	the columns to display in the tooltip.
<code>flag</code>	the special flag to indicate special plot.
<code>info</code>	additional information for plotting.
<code>...</code>	additional arguments.

Value

A chart object containing the information to create an interactive bar plot.

Author(s)

Shian Su

Examples

```
data(mtcars)
counts <- table(mtcars$gear)
data <- data.frame(nGears=as.numeric(names(counts)), Count=as.numeric(counts))

plot1 <- glBar(data, "Count", "nGears", ylab="Number of Gears")
glimma(plot1, layout=c(1,1), launch=TRUE)
```

glimma

Glimma plot manager

Description

Core glimma plot manager. Generates environment for glimma plots.

Usage

```
glimma(..., layout = c(1, 1), path = getwd(), folder = "glimma-plots",
  html = "index", overwrite = TRUE, launch = TRUE)
```

Arguments

...	the jschart or jslink objects for processing.
layout	the numeric vector representing the number of rows and columns in plot window.
path	the path in which the folder will be created.
folder	the name of the fold to save html file to.
html	the name of the html file to save plots to.
overwrite	the option to overwrite existing folder if it already exists.
launch	TRUE to launch plot after call.

Value

Generates interactive plots based on filling layout row by row from left to right.

Examples

```
data(iris)

plot1 <- glScatter(iris, xval="Sepal.Length", yval="Sepal.Width", colval="Species")
glimma(plot1, c(1,1))
```

glink

Plot linkages

Description

Helper function for writing the link properties in interactive Glimma plots

Usage

```
glink(from, to, src = "none", dest = "none", flag = "none",
      both = FALSE, info = "none")
```

Arguments

from	the index of the plot from which the event is dispatched.
to	the index of the plot which receives the event and performs an action.
src	the action that is performed in the "from" plot.
dest	the action that is performed in the "to" plot.
flag	indicates special links for particular chart types.
both	creates symmetric links whereby the "dest" action in "to" also triggers the "src" action in "from".
info	additional info for creating the link.

Value

a link object containing the plot linking information.

Examples

```
data(iris)
data <- data.frame(Name=paste("Flower", 1:nrow(iris), sep="-"), iris)

plot1 <- glScatter(data, xval="Sepal.Length", yval="Sepal.Width", colval="Species")
plot2 <- glScatter(data, xval="Species", yval="Petal.Length", colval="Species")
link1 <- glink(1, 2, src="hover", dest="hover", both=TRUE)
glimma(plot1, plot2, link1, layout=c(1,2))
```

`glMDPlot`*Glimma MD Plot*

Description

Draw an interactive MD plot

Usage

```
glMDPlot(x, ...)
```

Arguments

`x` the data.frame containing data to plot.
`...` additional arguments affecting the plots produced. See specific methods for detailed arguments.

Value

Draws a two-panel interactive MD plot in an html page. The left plot shows the log-fold-change vs average expression. The right plot shows the expression levels of a particular gene of each sample.

Author(s)

Shian Su

See Also

[glMDPlot.default](#), [glMDPlot.DGELRT](#), [glMDPlot.DGEEexact](#), [glMDPlot.MArrayLM](#), [glMDPlot.DESeqDataSet](#)

Examples

```
library(limma)
library(edgeR)

data(lymphomaRNAseq)
x <- lymphomaRNAseq

sel <- rowSums(cpm(x$counts)>0.5)>=3
x <- x[sel,]

genotype <- relevel(x$samples$group, "Smchd1-null")
x <- calcNormFactors(x, method="TMM")
des <- model.matrix(~genotype)

## Apply voom with sample quality weights and fit linear model
v <- voomWithQualityWeights(x, design=des, normalization="none", plot=FALSE)
vfit <- lmFit(v,des)
```

```
## Apply treat relative to a fold-change of 1.5
vtfits <- treat(vfit,lfc=log2(1.5))
vfit <- eBayes(vfit)
results <- decideTests(vfit,p.value=0.01)

glMDPlot(vfit, counts=x$counts, anno=x$genes, groups=genotype, samples=1:7,
          status=results[,2], main="MD plot: Wild-type vs Smchd1",
          display.columns=c("Symbols", "GeneID", "GeneName"),
          folder="Smchd1-Lymphoma")
```

glMDPlot.default

Glimma MD Plot

Description

Draw an interactive MD plot from a data.frame

Usage

```
## Default S3 method:
glMDPlot(x, xval, yval, counts, anno, groups, samples,
          status = rep(0, nrow(x)), xlab = xval, ylab = yval,
          side.xlab = "Group", side.ylab = "logCPM", search.by = "Symbols",
          jitter = 30, id.column = "GeneID", display.columns = id.column,
          cols = c("#0000FF", "#858585", "#B32222"), sample.cols = rep("#1f77b4",
          ncol(counts)), path = getwd(), folder = "glimma-plots",
          html = "MD-Plot", launch = TRUE, ...)
```

Arguments

x	the data.frame object containing expression and fold change values.
xval	the column to plot on x axis of left plot.
yval	the column to plot on y axis of left plot.
counts	the matrix containing all counts.
anno	the data.frame containing gene annotations.
groups	the factor containing experimental groups of the samples.
samples	the names of the samples.
status	vector giving the control status of data point, of same length as the number of rows of object. If NULL, then all points are plotted in the default colour.
xlab	the label on the x axis for the left plot.
ylab	the label on the y axis for the left plot.
side.xlab	the label on the x axis for the right plot.

side.ylab	the label on the y axis for the right plot.
search.by	the name of the column which will be used to search for data points. (should contain unique values)
jitter	the amount of jitter to apply to the samples in the expressions plot.
id.column	the column containing unique identifiers for each gene.
display.columns	character vector containing names of columns to display in mouseover tooltips.
cols	vector of strings denoting colours corresponding to control status -1, 0 and 1. (may be R named colours or Hex values)
sample.cols	vector of strings denoting colours for each sample point on the expression plot.
path	the path in which the folder will be created.
folder	the name of the fold to save html file to.
html	the name of the html file to save plots to.
launch	TRUE to launch plot after call.
...	additional arguments to be passed onto the MD plot.

Value

Draws a two-panel interactive MD plot in an html page. The left plot shows the log-fold-change vs average expression. The right plot shows the expression levels of a particular gene of each sample.

Author(s)

Shian Su

glMDPlot.DESeqDataSet *Glimma MD Plot*

Description

Draw an interactive MD plot from a DESeqDataSet object

Usage

```
## S3 method for class 'DESeqDataSet'
glMDPlot(x, anno, groups, samples, status = rep(0,
  nrow(x)), search.by = "Symbols", jitter = 30, id.column = "GeneID",
  display.columns = id.column, cols = c("#0000FF", "#858585", "#B32222"),
  sample.cols = rep("#1f77b4", ncol(x)), path = getwd(),
  folder = "glimma-plots", html = "MD-Plot", launch = TRUE, ...)
```

Arguments

x	the DESeqDataSet object.
anno	the data.frame containing gene annotations.
groups	the factor containing experimental groups of the samples.
samples	the names of the samples.
status	vector giving the control status of data point, of same length as the number of rows of object. If NULL, then all points are plotted in the default colour.
search.by	the name of the column which will be used to search for data points. (should contain unique values)
jitter	the amount of jitter to apply to the samples in the expressions plot.
id.column	the column containing unique identifiers for each gene.
display.columns	character vector containing names of columns to display in mouseover tooltips.
cols	vector of strings denoting colours corresponding to control status -1, 0 and 1. (may be R named colours or Hex values)
sample.cols	vector of strings denoting colours for each sample point on the expression plot.
path	the path in which the folder will be created.
folder	the name of the fold to save html file to.
html	the name of the html file to save plots to.
launch	TRUE to launch plot after call.
...	additional arguments to be passed onto the MD plot.

Value

Draws a two-panel interactive MD plot in an html page. The left plot shows the log-fold-change vs average expression. The right plot shows the expression levels of a particular gene of each sample.

Author(s)

Shian Su

glMDPlot.DESeqResults *Glimma MD Plot*

Description

Draw an interactive MD plot from a DESeqResults object

Usage

```
## S3 method for class 'DESeqResults'
gIMDPlot(x, counts, anno, groups, samples,
  status = rep(0, nrow(x)), search.by = "Symbols", jitter = 30,
  id.column = "GeneID", display.columns = id.column, cols = c("#0000FF",
  "#858585", "#B32222"), sample.cols = rep("#1f77b4", ncol(counts)),
  path = getwd(), folder = "glimma-plots", html = "MD-Plot",
  launch = TRUE, ...)
```

Arguments

x	the DESeqResults object.
counts	the matrix containing all counts.
anno	the data.frame containing gene annotations.
groups	the factor containing experimental groups of the samples.
samples	the names of the samples.
status	vector giving the control status of data point, of same length as the number of rows of object. If NULL, then all points are plotted in the default colour.
search.by	the name of the column which will be used to search for data points. (should contain unique values)
jitter	the amount of jitter to apply to the samples in the expressions plot.
id.column	the column containing unique identifiers for each gene.
display.columns	character vector containing names of columns to display in mouseover tooltips.
cols	vector of strings denoting colours corresponding to control status -1, 0 and 1. (may be R named colours or Hex values)
sample.cols	vector of strings denoting colours for each sample point on the expression plot.
path	the path in which the folder will be created.
folder	the name of the fold to save html file to.
html	the name of the html file to save plots to.
launch	TRUE to launch plot after call.
...	additional arguments to be passed onto the MD plot.

Value

Draws a two-panel interactive MD plot in an html page. The left plot shows the log-fold-change vs average expression. The right plot shows the expression levels of a particular gene of each sample.

Author(s)

Shian Su

glMDPlot.DGEEExact *Glimma MD Plot*

Description

Draw an interactive MD plot from a DGELRT objet

Usage

```
## S3 method for class 'DGEEExact'
glMDPlot(x, counts, anno, groups, samples, status = rep(0,
  nrow(x)), coef = ncol(x$coefficients), p.adj.method = "BH",
  search.by = "Symbols", jitter = 30, id.column = "GeneID",
  display.columns = id.column, cols = c("#0000FF", "#858585", "#B32222"),
  sample.cols = rep("#1f77b4", ncol(counts)), path = getwd(),
  folder = "glimma-plots", html = "MD-Plot", launch = TRUE, ...)
```

Arguments

x	the DGEEExact object.
counts	the matrix containing all counts.
anno	the data.frame containing gene annotations.
groups	the factor containing experimental groups of the samples.
samples	the names of the samples.
status	vector giving the control status of data point, of same length as the number of rows of object. If NULL, then all points are plotted in the default colour.
coef	integer or character index vector indicating which column of object to plot.
p.adj.method	character vector indicating multiple testing correction method. (defaults to "BH")
search.by	the name of the column which will be used to search for data points. (should contain unique values)
jitter	the amount of jitter to apply to the samples in the expressions plot.
id.column	the column containing unique identifiers for each gene.
display.columns	character vector containing names of columns to display in mouseover tooltips.
cols	vector of strings denoting colours corresponding to control status -1, 0 and 1. (may be R named colours or Hex values)
sample.cols	vector of strings denoting colours for each sample point on the expression plot.
path	the path in which the folder will be created.
folder	the name of the fold to save html file to.
html	the name of the html file to save plots to.
launch	TRUE to launch plot after call.
...	additional arguments to be passed onto the MD plot.

Value

Draws a two-panel interactive MD plot in an html page. The left plot shows the log-fold-change vs average expression. The right plot shows the expression levels of a particular gene of each sample.

Author(s)

Shian Su

glMDPlot.DGELRT

Glimma MD Plot

Description

Draw an interactive MD plot from a DGELRT object

Usage

```
## S3 method for class 'DGELRT'
glMDPlot(x, counts, anno, groups, samples, status = rep(0,
  nrow(x)), coef = ncol(x$coefficients), p.adj.method = "BH",
  search.by = "Symbols", jitter = 30, id.column = "GeneID",
  display.columns = id.column, cols = c("#0000FF", "#858585", "#B32222"),
  sample.cols = rep("#1f77b4", ncol(counts)), path = getwd(),
  folder = "glimma-plots", html = "MD-Plot", launch = TRUE, ...)
```

Arguments

x	the DGELRT object.
counts	the matrix containing all counts.
anno	the data.frame containing gene annotations.
groups	the factor containing experimental groups of the samples.
samples	the names of the samples.
status	vector giving the control status of data point, of same length as the number of rows of object. If NULL, then all points are plotted in the default colour.
coef	integer or character index vector indicating which column of object to plot.
p.adj.method	character vector indicating multiple testing correction method. (defaults to "BH")
search.by	the name of the column which will be used to search for data points. (should contain unique values)
jitter	the amount of jitter to apply to the samples in the expressions plot.
id.column	the column containing unique identifiers for each gene.
display.columns	character vector containing names of columns to display in mouseover tooltips.
cols	vector of strings denoting colours corresponding to control status -1, 0 and 1. (may be R named colours or Hex values)

sample.cols	vector of strings denoting colours for each sample point on the expression plot.
path	the path in which the folder will be created.
folder	the name of the fold to save html file to.
html	the name of the html file to save plots to.
launch	TRUE to launch plot after call.
...	additional arguments to be passed onto the MD plot.

Value

Draws a two-panel interactive MD plot in an html page. The left plot shows the log-fold-change vs average expression. The right plot shows the expression levels of a particular gene of each sample.

Author(s)

Shian Su

gIMDPlot.MArrayLM *Glimma MD Plot*

Description

Draw an interactive MD plot from a MArrayLM object

Usage

```
## S3 method for class 'MArrayLM'
gIMDPlot(x, counts, anno, groups, samples, status = rep(0,
  nrow(x)), coef = ncol(x$coefficients), p.adj.method = "BH",
  search.by = "Symbols", jitter = 30, id.column = "GeneID",
  display.columns = id.column, cols = c("#0000FF", "#858585", "#B32222"),
  sample.cols = rep("#1f77b4", ncol(counts)), path = getwd(),
  folder = "glimma-plots", html = "MD-Plot", launch = TRUE, ...)
```

Arguments

x	the MArrayLM object.
counts	the matrix containing all counts.
anno	the data.frame containing gene annotations.
groups	the factor containing experimental groups of the samples.
samples	the names of the samples.
status	vector giving the control status of data point, of same length as the number of rows of object. If NULL, then all points are plotted in the default colour.
coef	integer or character index vector indicating which column of object to plot.
p.adj.method	character vector indicating multiple testing correction method. (defaults to "BH")

search.by	the name of the column which will be used to search for data points. (should contain unique values)
jitter	the amount of jitter to apply to the samples in the expressions plot.
id.column	the column containing unique identifiers for each gene.
display.columns	character vector containing names of columns to display in mouseover tooltips.
cols	vector of strings denoting colours corresponding to control status -1, 0 and 1. (may be R named colours or Hex values)
sample.cols	vector of strings denoting colours for each sample point on the expression plot.
path	the path in which the folder will be created.
folder	the name of the fold to save html file to.
html	the name of the html file to save plots to.
launch	TRUE to launch plot after call.
...	additional arguments to be passed onto the MD plot.

Value

Draws a two-panel interactive MD plot in an html page. The left plot shows the log-fold-change vs average expression. The right plot shows the expression levels of a particular gene of each sample.

Author(s)

Shian Su

Examples

```
library(limma)
library(edgeR)

data(lymphomaRNAseq)
x <- lymphomaRNAseq

sel <- rowSums(cpm(x$counts)>0.5)>=3
x <- x[sel,]

genotype <- relevel(x$samples$group, "Smchd1-null")
x <- calcNormFactors(x, method="TMM")
des <- model.matrix(~genotype)

## Apply voom with sample quality weights and fit linear model
v <- voomWithQualityWeights(x, design=des, normalization="none", plot=FALSE)
vfit <- lmFit(v,des)

## Apply treat relative to a fold-change of 1.5
vtfits <- treat(vfit,lfc=log2(1.5))
vfit <- eBayes(vfits)
results <- decideTests(vfit,p.value=0.01)
```

```
gIMDSPlot(vfit, counts=x$counts, anno=x$genes, groups=genotype, samples=1:7,
          status=results[,2], main="MD plot: Wild-type vs Smchd1",
          display.columns=c("Symbols", "GeneID", "GeneName"),
          folder="Smchd1-Lymphoma")
```

gIMDSPlot

Glimma MDS Plot

Description

Draw an interactive MDS plot gene expression matrix with distances calculated from most variable genes.

Usage

```
gIMDSPlot(x, ...)
```

Arguments

`x` the data.frame containing data to plot.
`...` additional arguments affecting the plots produced. See specific methods for detailed arguments.

Value

Draws a two-panel interactive MDS plot in an html page. The left panel contains the plot between two MDS dimensions, with annotations displayed on hover. The right panel contains a bar plot of the eigenvalues of each dimension, clicking on any of the bars will plot the corresponding dimension against the next dimension.

Author(s)

Shian Su, Gordon Smyth

See Also

[gIMDSPlot.default](#), [gIMDSPlot.DGEList](#)

Examples

```
data(lymphomaRNAseq)
genotype <- relevel(lymphomaRNAseq$samples$group, "Smchd1-null")

gIMDSPlot(lymphomaRNAseq, labels=1:7, groups=genotype)
```

glMDSPlot.default *Glimma MDS Plot*

Description

Draw an interactive MDS plot from a gene expression matrix with distances calculated from most variable genes.

Usage

```
## Default S3 method:
glMDSPlot(x, top = 500, labels = 1:ncol(x),
  groups = rep(1, ncol(x)), gene.selection = "pairwise",
  main = "MDS Plot", path = getwd(), folder = "glimma-plots",
  html = "MDS-Plot", launch = TRUE, ...)
```

Arguments

x	the matrix containing the gene expressions.
top	the number of top most variable genes to use.
labels	the labels for each sample.
groups	the experimental group to which samples belong.
gene.selection	"pairwise" if most variable genes are to be chosen for each pair of samples or "common" to select the same genes for all comparisons.
main	the title of the plot.
path	the path in which the folder will be created.
folder	the name of the fold to save html file to.
html	the name of the html file to save plots to.
launch	TRUE to launch plot after call.
...	additional arguments.

Value

Draws a two-panel interactive MDS plot in an html page. The left panel contains the plot between two MDS dimensions, with annotations displayed on hover. The right panel contains a bar plot of the eigenvalues of each dimension, clicking on any of the bars will plot the corresponding dimension against the next dimension.

Author(s)

Shian Su, Gordon Smyth

gIMDSPlot.DGEList *Glimma MDS Plot*

Description

Draw an interactive MD plot from a DGEList object with distances calculated from most variable genes.

Usage

```
## S3 method for class 'DGEList'
gIMDSPlot(x, top = 500, labels = 1:ncol(x),
  groups = rep(1, ncol(x)), gene.selection = "pairwise",
  main = "MDS Plot", path = getwd(), folder = "glimma-plots",
  html = "MDS-Plot", launch = TRUE, ...)
```

Arguments

x	the DGEList containing the gene expressions.
top	the number of top most variable genes to use.
labels	the labels for each sample.
groups	the experimental group to which samples belong.
gene.selection	"pairwise" if most variable genes are to be chosen for each pair of samples or "common" to select the same genes for all comparisons.
main	the title of the plot.
path	the path in which the folder will be created.
folder	the name of the fold to save html file to.
html	the name of the html file to save plots to.
launch	TRUE to launch plot after call.
...	additional arguments.

Value

Draws a two-panel interactive MDS plot in an html page. The left panel contains the plot between two MDS dimensions, with annotations displayed on hover. The right panel contains a bar plot of the eigenvalues of each dimension, clicking on any of the bars will plot the corresponding dimension against the next dimension.

Author(s)

Shian Su, Gordon Smyth

glScatter *Glimma Scatter Plot*

Description

Create an interactive scatter plot object

Usage

```
glScatter(x, ...)
```

Arguments

x the data.frame containing data to plot.
... additional arguments depending on input object type.

Value

A chart object containing the information to create an interactive scatter plot.

Author(s)

Shian Su

Examples

```
data(iris)  
  
plot1 <- glScatter(iris, xval="Sepal.Length", yval="Sepal.Width", colval="Species")  
glimma(plot1, c(1,1))
```

glScatter.default *Glimma Scatter Plot*

Description

Default method for creating an interactive scatter plot

Usage

```
## Default S3 method:  
glScatter(x, xval = "x", yval = "y", idval = NULL,  
          x.jitter = 0, y.jitter = 0, ndigits = NULL, signif = 6, log = "",  
          xlab = xval, ylab = yval, main = NULL, height = 400, width = 500,  
          colval = NULL, annot = c(xval, yval), annot.lab = NULL, flag = NULL,  
          info = NULL, hide = FALSE, ...)
```

Arguments

<code>x</code>	the data.frame containing data to plot.
<code>xval</code>	the column name for the x-axis values.
<code>yval</code>	the column name for the y-axis values.
<code>idval</code>	the column name for unique identifiers.
<code>x.jitter</code>	the amount of jittering to add to values along the x axis.
<code>y.jitter</code>	the amount of jittering to add to values along the y axis.
<code>ndigits</code>	the number of digits after the decimal to round to in the tooltip (overrides <code>signif</code>).
<code>signif</code>	the number of significant figures to display in the tooltip.
<code>log</code>	a character string which contains "x" if the x axis is to be logarithmic, "y" if the y axis is to be logarithmic and "xy" or "yx" if both axes are to be logarithmic.
<code>xlab</code>	the label on the x-axis.
<code>ylab</code>	the label on the y-axis.
<code>main</code>	the title for the plot.
<code>height</code>	the height of the plot (in pixels).
<code>width</code>	the width of the plot (in pixels).
<code>colval</code>	the colours for each data point.
<code>annot</code>	the columns to display in the tooltip.
<code>annot.lab</code>	alternative labels for the values displayed in the tooltip.
<code>flag</code>	the special flag to indicate special plot.
<code>info</code>	additional information for plotting.
<code>hide</code>	TRUE to hide the plot when page starts.
<code>...</code>	additional arguments.

Value

A chart object containing the information to create an interactive scatter plot.

Author(s)

Shian Su

Examples

```
data(iris)
```

```
plot1 <- glScatter(iris, xval="Sepal.Length", yval="Sepal.Width", colval="Species")  
glimma(plot1, c(1,1))
```

hasColumns	<i>Column checker</i>
------------	-----------------------

Description

Check if data.frame controls all the listed columns

Usage

```
hasColumns(df, columns)
```

Arguments

df	the data frame to check.
columns	the columns that should exist in the data frame.

Value

stops program with an error if column cannot be found in df

lymphomaRNAseq	<i>Mouse based RNAseq data for study of smchd1 gene.</i>
----------------	--

Description

Mouse based RNAseq data for study of smchd1 gene.

Author(s)

Ruijie Liu, Kelan Chen, Natasha Jansz, Marnie E. Blewitt, Matthew E. Ritchie

References

<http://www.sciencedirect.com/science/article/pii/S2213596015301306>

makeChartJson

JSON converter for chart objects

Description

Function to make json object from a chart, ignoring the json property

Usage

```
makeChartJson(chart)
```

Arguments

chart the chart object to be converted into JSON

Value

a stringified JSON object containing the chart data.

makeDFJson

JSON converter for data frames

Description

Function to create a JSON from a data.frame

Usage

```
makeDFJson(df)
```

Arguments

df the data.frame to be converted into JSON

Value

a stringified JSON, the data.frame is encoded as a vector of objects, with each column being one object with keys corresponding to column names.

NumToHexCol

Numeric to hex colour converter

Description

Functions to convert numbers into corresponding hex codes for colours

Usage

NumToHexCol(x)

Arguments

x the colour value(s) to be converted to hex values.

Value

hex codes for colours

Index

*Topic **RNAseq**

lymphomaRNAseq, [21](#)

CharToHexCol, [2](#)

glAutoinput, [3](#)

glBar, [3](#)

glBar.default, [4](#), [4](#)

glimma, [5](#)

gllink, [6](#)

glMDPlot, [7](#)

glMDPlot.default, [7](#), [8](#)

glMDPlot.DESeqDataSet, [7](#), [9](#)

glMDPlot.DESeqResults, [10](#)

glMDPlot.DGEEExact, [7](#), [12](#)

glMDPlot.DGELRT, [7](#), [13](#)

glMDPlot.MArrayLM, [7](#), [14](#)

glMDSPlot, [16](#)

glMDSPlot.default, [16](#), [17](#)

glMDSPlot.DGEList, [16](#), [18](#)

glScatter, [19](#)

glScatter.default, [19](#)

hasColumns, [21](#)

lymphomaRNAseq, [21](#)

makeChartJson, [22](#)

makeDFJson, [22](#)

NumToHexCol, [23](#)