Package 'debrowser'

April 14, 2017

Type Package

Title debrowser: Interactive Differential Expresion Analysis Browser

Version 1.3.11

Date 2016-07-05

Author Alper Kucukural <`alper.kucukural@umassmed.edu`>,
Manuel Garber <`manuel.garber@umassmed.edu`>

Maintainer Alper Kucukural <`alper.kucukural@umassmed.edu`>

Description Bioinformatics platform containing interactive plots and tables for differential gene and region expression studies. Allows visualizing expression data much more deeply in an interactive and faster way. By changing the parameters, users can easily discover different parts of the data that like never have been done before. Manually creating and looking these plots takes time. With DEBrowser users can prepare plots without writing any code. Differential expression, PCA and clustering analysis are made on site and the results are shown in various plots such as scatter, bar, box, volcano, ma plots and Heatmaps.

Depends R (>= 3.3.0), shiny, ggvis, jsonlite, shinyjs

License GPL-3 + file LICENSE

LazyData true

Imports DT, ggplot2, RColorBrewer, annotate, gplots, AnnotationDbi, DESeq2, DOSE, igraph, grDevices, graphics, stats, utils, GenomicRanges, IRanges, S4Vectors, SummarizedExperiment, stringi, reshape2, baySeq, d3heatmap, org.Hs.eg.db, org.Mm.eg.db, limma, edgeR, clusterProfiler, V8, methods, sva, shinydashboard, devtools, RCurl

RoxygenNote 6.0.1

Suggests testthat, rmarkdown, knitr, R.rsp

VignetteBuilder knitr, R.rsp

URL `https://github.com/UMMS-Biocore/debrowser`

BugReports `https://github.com/UMMS-Biocore/debrowser/issues/new`

biocViews Sequencing, ChIPSeq, RNASeq, DifferentialExpression, GeneExpression, Clustering

NeedsCompilation no

R topics documented:

`actionButton` *Buttons including Action Buttons and Event Buttons*

Description

Creates an action button whose value is initially zero, and increments by one each time it is pressed.

Usage

```
actionButton(inputId, label, styleclass = "", size = "", block = FALSE,
  icon = NULL, css.class = "", ...)
```

Arguments

`inputId`	Specifies the input slot that will be used to access the value.
`label`	The contents of the button–usually a text label, but you could also use any other HTML, like an image.
`styleclass`	The Bootstrap styling class of the button–options are primary, info, success, warning, danger, inverse, link or blank
`size`	The size of the button–options are large, small, mini
`block`	Whehter the button should fill the block
`icon`	Display an icon for the button
`css.class`	Any additional CSS class one wishes to add to the action button
`...`	Other argument to feed into shiny::actionButton

Examples

```
actionButton("goDE", "Go to DE Analysis!")
```

`addDataCols` *addDataCols*

Description

add aditional data columns to de results

Usage

```
addDataCols(data = NULL, de_res = NULL, cols = NULL, inputconds = NULL,
  i = NULL, input = NULL)
```

Arguments

`data,`	loaded dataset
`de_res,`	de results
`cols,`	columns
`inputconds,`	inputconds
`i,`	selected comparison number
`input,`	input

Value

data

Examples

```
x <- addDataCols()
```

`addID` *addID*

Description

Adds an id to the data frame being used.

Usage

```
addID(data = NULL)
```

Arguments

`data,`	loaded dataset

Value

data

Examples

```
x <- addID()
```

---

add_title_pos *add_title_pos*

Description

Adds a title with extra axis to ggvis plot and sets the positions

Usage

```
add_title_pos(vis, ..., title = "Plot Title", align = "left", angle = 0,
  dx = 0, dy = 0)
```

Arguments

`vis,`	a ggvis plot
`...,`	any additional arguments
`title`	for the plot
`align`	position of the title c('left','right')
`angle`	of the labels in x axis
`dx,`	relative x position of the labels in the x axis
`dy,`	relative y position of the labels in the x axis

Value

deseq2 results

Examples

```
require(ggvis)
mtcars %>%
ggvis(x=~cyl, y=~wt, fill=~mpg) %>%
group_by(mpg) %>%
layer_bars() %>%
add_title_pos(title = "title", angle=310, dy=0, dx=0) %>%
set_options(width = 400, height = 350)
```

`all2all` *all2all*

Description

Prepares all2all scatter plots for given datasets.

Usage

```
all2all(data, cex = 2)
```

Arguments

`data,`	data that have the sample names in the header.
`cex`	text size

Value

all2all scatter plots

Examples

```
plot<-all2all(mtcars)
```

`applyFilters` *applyFilters*

Description

Applies filters based on user selected parameters to be displayed within the DEBrowser.

Usage

```
applyFilters(filt_data = NULL, cols = NULL, conds = NULL, input = NULL)
```

Arguments

`filt_data,`	loaded dataset
`cols,`	selected samples
`conds,`	seleced conditions
`input,`	input parameters

Value

data

Examples

```
x <- applyFilters()
```

`applyFiltersToMergedComparison` *applyFiltersToMergedComparison*

Description

Gathers the merged comparison data to be used within the DEBrowser.

Usage

```
applyFiltersToMergedComparison(merged = NULL, nc = NULL, input = NULL)
```

Arguments

`merged,`	merged data
`nc,`	the number of comparisons
`input,`	input params

Value

data

Examples

```
x <- applyFiltersToMergedComparison()
```

`bookmarkServer` *bookmarkServer*

Description

bookmark Server functions

Usage

```
bookmarkServer(input = NULL, output = NULL, session = NULL,
  loadingJSON = NUL)
```

Arguments

`input,`	input
`output,`	output
`session,`	session
`loadingJSON,`	loadingJSON

Examples

```
x <- bookmarkServer()
```

`bookmarkUI` *bookmarkUI*

Description

bookmark UI

Usage

```
bookmarkUI(id = NULL)
```

Arguments

`id,`	id

Examples

```
x <- bookmarkUI()
```

`clusterData` *clusterData*

Description

Gathers the Cluster analysis data to be used within the GO Term plots.

Usage

```
clusterData(dat = NULL)
```

Arguments

`dat,`	the data to cluster

Value

clustered data

Note

`clusterData`

Examples

```
mycluster <- clusterData()
```

`compareClust` *compareClust*

Description

Compares the clustered data to be displayed within the GO Term plots.

Usage

```
compareClust(dat = NULL, ont = "CC", org = "org.Hs.eg.db",
  fun = "enrichGO", title = "Ontology Distribution Comparison",
  pvalueCutoff = 0.01)
```

Arguments

`dat,`	data to compare clusters
`ont,`	the ontology to use
`org,`	the organism used
`fun,`	fun
`title,`	title of the comparison
`pvalueCutoff,`	pvalueCutoff

Value

compared cluster

Note

`compareClust`

Examples

```
    x <- compareClust()
```

`copy2newDirectory` *copy2newDirectory*

Description

To copy the bookmarked folder into a user named directory

Usage

```
copy2newDirectory(new_state_id = NULL, username = NULL, session = NULL)
```

Arguments

`new_state_id,`	new state id
`username,`	username
`session,`	session

Examples

```
x <- copy2newDirectory()
```

`correctBatchEffect` *Correct Batch Effect*

Description

Batch effect correction

Usage

```
correctBatchEffect(idata = NULL, input = NULL)
```

Arguments

`idata,`	data
`input,`	input values

Value

data

Examples

```
x<-correctBatchEffect ()
```

`deServer` *deServer*

Description

Sets up shinyServer to be able to run DEBrowser interactively.

Usage

```
deServer(input, output, session)
```

Arguments

`input,`	input params from UI
`output,`	output params to UI
`session,`	session variable

Value

the panel for main plots;

Note

`deServer`

Examples

```
deServer
```

`deUI` *deUI*

Description

Creates a shinyUI to be able to run DEBrowser interactively.

Usage

```
deUI()
```

Value

the panel for main plots;

Note

`deUI`

Examples

```
x<-deUI()
```

`drawPCAExplained` *drawPCAExplained*

Description

Creates a more detailed plot using the PCA results from the selected dataset.

Usage

```
drawPCAExplained(explainedData = NULL)
```

Arguments

`explainedData,`	selected data

Value

explained plot

Examples

```
x <- drawPCAExplained()
```

`getAfterLoadMsg` *getAfterLoadMsg*

Description

Generates and displays the message to be shown after loading data within the DEBrowser.

Usage

```
getAfterLoadMsg()
```

Value

return After Load Msg

Note

`getAfterLoadMsg`

Examples

```
x <- getAfterLoadMsg()
```

`getColors` *getColors*

Description

get colors for the domains

Usage

```
getColors(domains = NULL)
```

Arguments

`domains,`	domains to be colored

Value

colors

Examples

```
x<-getColors()
```

`getColorShapeSelection` *getColorShapeSelection*

Description

Generates the fill and shape selection boxes for PCA plots. metadata file has to be loaded in this case

Usage

```
getColorShapeSelection(input = NULL)
```

Arguments

`input,`	input values

Value

Color and shape selection boxes

Examples

```
x <- getColorShapeSelection()
```

`getCompSelection` *getCompSelection*

Description

Gathers the user selected comparison set to be used within the DEBrowser.

Usage

```
getCompSelection(count = NULL)
```

Arguments

`count,`	comparison count

Note

`getCompSelection`

Examples

```
x <- getCompSelection(count = 2)
```

`getConditionSelector` *getConditionSelector*

Description

Selects user input conditions to run in DESeq.

Usage

```
getConditionSelector(num = 0, choices = NULL, selected = NULL)
```

Arguments

`num,`	panel that is going to be shown
`choices,`	sample list
`selected,`	selected smaple list

Examples

```
    x <- getConditionSelector()
```

`getConditionSelectorFromMeta` *getConditionSelectorFromMeta*

Description

Selects user input conditions to run in DESeq from metadata

Usage

```
getConditionSelectorFromMeta(input = NULL, index = 1, num = 0,
  choices = NULL, selected = NULL, loadingJSON = NULL)
```

Arguments

`input,`	input
`index,`	index
`num,`	num
`choices,`	choices
`selected,`	selected
`loadingJSON,`	loadingJSON

Examples

```
    x <- getConditionSelectorFromMeta()
```

`getCondMsg` *getCondMsg*

Description

Generates and displays the current conditions and their samples within the DEBrowser.

Usage

```
getCondMsg(dc = NULL, num = NULL, cols = NULL, conds = NULL)
```

Arguments

`dc,`	columns
`num,`	selected comparison
`cols,`	columns
`conds,`	selected conditions

Value

return conditions

Note

`getCondMsg`

Examples

```
x <- getCondMsg()
```

`getCutOffSelection` *getCutOffSelection*

Description

Gathers the cut off selection for DE analysis

Usage

```
getCutOffSelection(nc = 1)
```

Arguments

`nc,`	total number of comparisons

Value

returns the left menu according to the selected tab;

Note

`getCutOffSelection`

Examples

```
x <- getCutOffSelection()
```

`getDataForTables` *getDataForTables get data to fill up tables tab*

Description

getDataForTables get data to fill up tables tab

Usage

```
getDataForTables(input = NULL, init_data = NULL, filt_data = NULL,
  selected = NULL, getMostVaried = NULL, mergedComp = NULL,
  explainedData = NULL)
```

Arguments

`input,`	input parameters
`init_data,`	initial dataset
`filt_data,`	filt_data
`selected,`	selected genes
`getMostVaried,`	most varied genes
`mergedComp,`	merged comparison set
`explainedData,`	pca gene set

Value

data

Examples

```
x <- getDataForTables()
```

`getDataPrepPanel` *getDataPrepPanel*

Description

Create and show the Condition selection screen to the user within the DEBrowser.

Usage

```
getDataPrepPanel(flag = FALSE)
```

Arguments

`flag,`	flag to show the element in the ui

Value

returns the left menu according to the selected tab;

Note

`getDataPrepPanel`

Examples

```
x <- getDataPrepPanel()
```

`getDensityPlot` *getDensityPlot*

Description

Makes Density plots

Usage

```
getDensityPlot(data = NULL, cols = NULL, title = "")
```

Arguments

`data,`	count or normalized data
`cols,`	columns
`title,`	title

Examples

```
getDensityPlot()
```

`getDomains` *getDomains*

Description

Get domains for the main plots.

Usage

```
getDomains(filt_data = NULL)
```

Arguments

`filt_data,`	data to get the domains

Value

domains

Examples

```
x<-getDomains()
```

`getDown` *getDown get down regulated data*

Description

getDown get down regulated data

Usage

```
getDown(filt_data = NULL)
```

Arguments

`filt_data,`	filt_data

Value

data

Examples

```
x <- getDown()
```

`getDownloadSection` *getDownloadSection*

Description

download section button and dataset selection box in the menu for user to download selected data.

Usage

```
getDownloadSection(flag = FALSE, choices = NULL)
```

Arguments

`flag,`	to show the download selection
`choices,`	main vs. QC section

Value

the panel for download section in the menu;

Note

`getDownloadSection`

Examples

```
x<- getDownloadSection()
```

`getEnrichDO` *getEnrichDO*

Description

Gathers the Enriched DO Term analysis data to be used within the GO Term plots.

Usage

```
getEnrichDO(genelist = NULL, pvalueCutoff = 0.01)
```

Arguments

`genelist,`	gene list
`pvalueCutoff,`	the p value cutoff

Value

enriched DO

Note

`getEnrichDO`

Examples

```
x <- getEnrichDO()
```

`getEnrichGO` *getEnrichGO*

Description

Gathers the Enriched GO Term analysis data to be used within the GO Term plots.

Usage

```
getEnrichGO(genelist = NULL, pvalueCutoff = 0.01, org = "org.Hs.eg.db",
  ont = "CC")
```

Arguments

`genelist,`	gene list
`pvalueCutoff,`	p value cutoff
`org,`	the organism used
`ont,`	the ontology used

Value

Enriched GO

Note

`getEnrichGO`

Examples

```
x <- getEnrichGO()
```

`getEnrichKEGG` *getEnrichKEGG*

Description

Gathers the Enriched KEGG analysis data to be used within the GO Term plots.

Usage

```
getEnrichKEGG(genelist = NULL, pvalueCutoff = 0.01, org = "org.Hs.eg.db")
```

Arguments

`genelist,`	gene list
`pvalueCutoff,`	the p value cutoff
`org,`	the organism used

Value

Enriched KEGG

Note

`getEnrichKEGG`

Examples

```
x <- getEnrichKEGG()
```

`getGeneList` *getGeneList*

Description

Gathers the gene list to use for GOTerm analysis.

Usage

```
getGeneList(genes = NULL, org = "org.Hs.eg.db")
```

Arguments

`genes,`	gene list
`org,`	orgranism for gene symbol entrez ID conversion

Value

ENTREZ ID list

Note

`GOTerm`

`getGeneList` symobol to ENTREZ ID conversion

Examples

```
x <- getGeneList(c('OCLN', 'ABCC2'))
```

`getGeneSetData` *getGeneSetData*

Description

Gathers the specified gene set list to be used within the DEBrowser.

Usage

```
getGeneSetData(data = NULL, geneset = NULL)
```

Arguments

`data,`	loaded dataset
`geneset,`	given gene set

Value

data

Examples

```
x <- getGeneSetData()
```

`getGOLeftMenu` *getGOLeftMenu*

Description

Generates the GO Left menu to be displayed within the DEBrowser.

Usage

```
getGOLeftMenu()
```

Value

returns the left menu according to the selected tab;

Note

`getGOLeftMenu`

Examples

```
x <- getGOLeftMenu()
```

`getGoPanel` *getGoPanel*

Description

Creates go term analysis panel within the shiny display.

Usage

```
getGoPanel(flag = FALSE)
```

Arguments

`flag,`	flag to show the element in the ui

Value

the panel for go term analysis;

Note

`getGoPanel`

Examples

```

x <- getGoPanel()
```

`getGOPlots` *getGOPlots*

Description

Go term analysis panel. Generates appropriate GO plot based on user selection.

Usage

```
getGOPlots(dataset = NULL, input = NULL)
```

Arguments

`dataset,`	the dataset used
`input,`	input params

Value

the panel for go plots;

Note

`getGOPlots`

Examples

```
x<- getGOPlots()
```

`getHelpButton` *getHelpButton prepares a helpbutton for to go to a specific site in the documentation*

Description

getHelpButton prepares a helpbutton for to go to a specific site in the documentation

Usage

```
getHelpButton(name = NULL, link = NULL)
```

Arguments

`name,`	name that are going to come after info
`link,`	link of the help

Value

the info button

Examples

```
x<- getHelpButton()
```

`getHoverPlots` *getHoverPlots*

Description

Prepares the plots going to be shown when a gene hovered in the main plots

Usage

```
getHoverPlots(bardata = NULL, genename = NULL)
```

Arguments

`bardata,`	barplot data
`genename,`	gene name in the barplots

Examples

```
getHoverPlots()
```

`getInitialMenu` *getInitialMenu*

Description

Displays the initial menu within DEBrowser.

Usage

```
getInitialMenu(input = NULL, output = NULL, session = NULL)
```

Arguments

`input,`	input from user
`output,`	output to user
`session,`	session info

Value

returns the initial menu

Note

`getInitialMenu`

Examples

```
x <- getInitialMenu()
```

`getIntHeatmap` *getIntHeatmap*

Description

getIntHeatmap

Usage

```
getIntHeatmap(data = NULL, input = NULL, inputQCPlot = NULL)
```

Arguments

`data,`	heatData
`input,`	all input params
`inputQCPlot,`	input poarams for QC

Value

plot

Examples

```
getIntHeatmap()
```

`getIntHeatmapVis` *getIntHeatmapVis*

Description

Gathers the conditional panel for interactive heatmap

Usage

```
getIntHeatmapVis(randstr = NULL)
```

Arguments

`randstr,`	randstr

Value

the panel interactive heatmap

Note

`getIntHeatmapVis`

Examples

```
x <- getIntHeatmapVis()
```

`getIQRPlot` *getIQRPlot*

Description

Makes IQR boxplot plot

Usage

```
getIQRPlot(data = NULL, cols = NULL, title = "")
```

Arguments

`data,`	count or normalized data
`cols,`	columns
`title,`	title

Examples

```
getIQRPlot()
```

`getJsonObj` *getJsonObj*

Description

getJsonVars

Usage

```
getJsonObj(session = NULL, input = NULL, access_token = NULL)
```

Arguments

`session,`	session
`input,`	input
`access_token,`	access_token

Examples

```
x <- getJsonObj()
```

`getLeftMenu` *getLeftMenu*

Description

Generates the left menu for for plots within the DEBrowser.

Usage

```
getLeftMenu(input = NULL)
```

Arguments

`input,`	input values

Value

returns the left menu according to the selected tab;

Note

`getLeftMenu`

Examples

```
x <- getLeftMenu()
```

`getLegendSelect` *getLegendSelect*

Description

select legend

Usage

```
getLegendSelect()
```

Note

`getLegendSelect`

Examples

```
x <- getLegendSelect()
```

`getLoadingMsg` *getLoadingMsg*

Description

Creates and displays the loading message/gif to be displayed within the DEBrowser.

Usage

```
getLoadingMsg(output = NULL)
```

Arguments

`output,`	output message

Value

loading msg

Note

`getLoadingMsg`

Examples

```
x <- getLoadingMsg()
```

`getLogo` *getLogo*

Description

Generates and displays the logo to be shown within DEBrowser.

Usage

```
getLogo()
```

Value

return logo

Note

`getLogo`

Examples

```
x <- getLogo()
```

getMainPanel *getMainPanel*

Description

main panel for volcano, scatter and maplot. Barplot and box plots are in this page as well.

Usage

```
getMainPanel(randstr = NULL)
```

Arguments

`randstr,`	random string for the plot containers

Value

the panel for main plots;

Note

getMainPanel

Examples

```
x <- getMainPanel()
```

getMainPanelPlots *getMainPanelPlots*

Description

Gathers the the plots to be used within the main panel.

Usage

```
getMainPanelPlots(filt_data = NULL, cols = NULL, conds = NULL,
  input = NULL, compselect = NULL)
```

Arguments

`filt_data,`	filtered data
`cols,`	selected columns
`conds,`	seleced conditions
`input,`	input from ui
`compselect,`	selected comparison number

Value

panel

Examples

```
x <- getMainPanelPlots()
```

`getMainPlotsLeftMenu` *getMainPlotsLeftMenu*

Description

Generates the Main PLots Left menu to be displayed within the DEBrowser.

Usage

```
getMainPlotsLeftMenu()
```

Value

returns the left menu according to the selected tab;

Note

`getMainPlotsLeftMenu`

Examples

```
x <- getMainPlotsLeftMenu()
```

`getMean` *getMean*

Description

Gathers the mean for selected condition.

Usage

```
getMean(norm_data = NULL, de_res = NULL, inputconds = NULL,
  colnum = NULL)
```

Arguments

`norm_data,`	loaded dataset
`de_res,`	de results
`inputconds,`	input parameters
`colnum,`	colnum

Value

data

Examples

```
x <- getMean()
```

`getMergedComparison` *getMergedComparison*

Description

Gathers the merged comparison data to be used within the DEBrowser.

Usage

```
getMergedComparison(Dataset = NULL, dc = NULL, nc = NULL, input = NULL)
```

Arguments

`Dataset,`	whole data
`dc,`	data container
`nc,`	the number of comparisons
`input,`	input params

Value

data

Examples

```
x <- getMergedComparison()
```

`getMethodDetails` *get the detail boxes after DE method selected*

Description

get the detail boxes after DE method selected

Usage

```
getMethodDetails(num = 0, input = NULL)
```

Arguments

`num,`	panel that is going to be shown
`input,`	user input

Examples

```
x <- getMethodDetails()
```

`getMostVariedList` *getMostVariedList*

Description

Calculates the most varied genes to be used for specific plots within the DEBrowser.

Usage

```
getMostVariedList(datavar = NULL, cols = NULL, input = NULL)
```

Arguments

`datavar,`	loaded dataset
`cols,`	selected columns
`input,`	input

Value

data

Examples

```
x <- getMostVariedList()
```

`getNormalizedMatrix` *getNormalizedMatrix*

Description

Normalizes the matrix passed to be used within various methods within DEBrowser. Requires edgeR package

Usage

```
getNormalizedMatrix(M = NULL, method = "TMM")
```

Arguments

`M,`	numeric matrix
`method,`	normalization method for edgeR. default is TMM

Value

normalized matrix

Note

`getGoPanel`

Examples

```
x <- getNormalizedMatrix(mtcars)
```

`getOrganism` *getOrganism*

Description

getOrganism

Usage

```
getOrganism(org)
```

Arguments

`org,`	organism

Value

organism name for keg

Note

`getOrganism`

Examples

```
x <- getOrganism()
```

`getOrganismBox` *getOrganismBox*

Description

Get the organism Box.

Usage

```
getOrganismBox()
```

Value

selectInput

Note

`getOrganismBox`

`getOrganismBox` makes the organism box

Examples

```
x <- getOrganismBox()
```

`getOrganismPathway` *getOrganismPathway*

Description

getOrganismPathway

Usage

```
getOrganismPathway(org)
```

Arguments

`org,`	organism

Value

organism name for pathway

Note

`getOrganismPathway`

Examples

```
x <- getOrganismPathway()
```

`getPCAexplained` *getPCAexplained*

Description

Creates a more detailed plot using the PCA results from the selected dataset.

Usage

```
getPCAexplained(datasetInput = NULL, input = NULL)
```

Arguments

`datasetInput,`	selected data
`input,`	from user

Value

explained plot

Examples

```
load(system.file("extdata", "demo", "demodata.Rda",
package="debrowser"))
input<-c()
input$qcplot<-"pca"
input$col_list<-colnames(demodata[,2:7])
x <- getPCAexplained(getNormalizedMatrix(demodata[,2:7]),
    input)
```

`getPCselection` *getPCselection*

Description

Generates the PC selection number to be used within DEBrowser.

Usage

```
getPCselection(num = 1, xy = "x")
```

Arguments

`num,`	PC selection number
`xy,`	x or y coordinate

Value

PC selection for PCA analysis

Note

`getPCselection`

Examples

```
    x <- getPCselection()
```

`getProgramTitle` *getProgramTitle*

Description

Generates the title of the program to be displayed within DEBrowser. If it is called in a program, the program title will be hidden

Usage

```
getProgramTitle(session = NULL)
```

Arguments

`session,`	session var

Value

program title

Note

`getProgramTitle`

Examples

```
title<-getProgramTitle()
```

`getQCLeftMenu` *getQCLeftMenu*

Description

Generates the left menu to be used for QC plots within the DEBrowser.

Usage

```
getQCLeftMenu(input = NULL)
```

Arguments

`input,`	input values

Value

QC left menu

Note

`getQCLeftMenu`

Examples

```
x <- getQCLeftMenu()
```

`getQCPanel` *getQCPanel*

Description

Gathers the conditional panel for QC plots

Usage

```
getQCPanel(input = NULL)
```

Arguments

`input,`	user input

Value

the panel for QC plots

Note

`getQCSection`

Examples

```
x <- getQCPanel()
```

`getQCPlots` *getQCPlots*

Description

Gathers the plot data to be displayed within the quality checks panel.

Usage

```
getQCPlots(dataset = NULL, input = NULL, metadata = NULL,
  inputQCPlot = NULL, drawPCAExplained = NULL)
```

Arguments

`dataset,`	the dataset to use
`input,`	user input
`metadata,`	coupled samples and conditions
`inputQCPlot,`	input QC params
`drawPCAExplained,`	to draw pca loading plot

Value

the panel for QC plots

Note

`getQCPlots`

Examples

```
x <- getQCPlots()
```

`getQCReplot` *getQCReplot*

Description

Prepares QCplots for comparisons and others

Usage

```
getQCReplot(cols = NULL, conds = NULL, datasetInput = NULL,
  input = NULL, inputQCPlot = NULL, drawPCAExplained = NULL)
```

Arguments

`cols,`	the dataset to use
`conds,`	the dataset to use
`datasetInput,`	the dataset to use
`input,`	user input
`inputQCPlot,`	input QC params
`drawPCAExplained,`	to draw pca loading plot

Value

the panel for QC plots

Note

`getQCReplot`

Examples

```
x <- getQCReplot()
```

`getSampleNames` *getSampleNames*

Description

Prepares initial samples to fill condition boxes. it reads the sample names from the data and splits into two.

Usage

```
getSampleNames(cnames = NULL, part = 1)
```

Arguments

`cnames,`	sample names in the header of a dataset
`part,`	c(1,2). 1=first half and 2= second half

Value

sample names.

Examples

```
x<-getSampleNames()
```

`getSamples` *getSamples*

Description

Gathers the sample names to be used within DEBrowser.

Usage

```
getSamples(cnames = NULL, index = 2)
```

Arguments

`cnames,`	names of the samples
`index,`	starting column in a tab separated file

Value

choices

Examples

```
x <- getSamples()
```

`getSearchData` *getSearchData*

Description

search the geneset in the tables and return it

Usage

```
getSearchData(dat = NULL, input = NULL)
```

Arguments

`dat,`	table data
`input,`	input params

Value

data

Examples

```
x <- getSearchData()
```

`getSelectedCols` *getSelectedCols*

Description

gets selected columns

Usage

```
getSelectedCols(data = NULL, datasetInput = NULL, input = NULL)
```

Arguments

`data,`	all loaded data
`datasetInput,`	selected dataset
`input,`	user input params

Examples

```
getSelectedCols()
```

`getSelectedDatasetInput` *getSelectedDatasetInput*

Description

Gathers the user selected dataset output to be displayed.

Usage

```
getSelectedDatasetInput(rdata = NULL, getSelected = NULL,
  getMostVaried = NULL, mergedComparison = NULL, explainedData = NULL,
  input = NULL)
```

Arguments

`rdata,`	filtered dataset
`getSelected,`	selected data
`getMostVaried,`	most varied data
`mergedComparison,`	merged comparison data
`explainedData,`	pca set
`input,`	input parameters

Value

data

Examples

```
x <- getSelectedDatasetInput()
```

`getSelectInputBox` *getSelectInputBox*

Description

Selects user input conditions to run in DESeq.

Usage

```
getSelectInputBox(id = NULL, name = NULL, num = 0, choices = NULL,
  selected = NULL, cw = 2)
```

Arguments

`id,`	input id
`name,`	label of the box
`num,`	panel that is going to be shown
`choices,`	sample list
`selected,`	selected smaple list
`cw,`	column width

Examples

```
x <- getSelectInputBox()
```

`getSelHeat` *getSelHeat*

Description

heatmap selection functionality

Usage

```
getSelHeat(data = NULL, input = NULL)
```

Arguments

`data,`	selected genes
`input,`	input params

Value

plot

Examples

```
x <- getSelHeat()
```

`getShapeColor` *getShapeColor*

Description

Generates the fill and shape selection boxes for PCA plots. metadata file has to be loaded in this case

Usage

```
getShapeColor(input = NULL)
```

Arguments

`input,`	input values

Value

Color and shape from selection boxes or defaults

Examples

```
x <- getShapeColor()
```

`getStartPlotsMsg` *getStartPlotsMsg*

Description

Generates and displays the starting messgae to be shown once the user has first seen the main plots page within DEBrowser.

Usage

```
getStartPlotsMsg()
```

Value

return start plot msg

Note

`getStartPlotsMsg`

Examples

```
x <- getStartPlotsMsg()
```

`getStartupMsg` *getStartupMsg*

Description

Generates and displays the starting message within DEBrowser.

Usage

```
getStartupMsg()
```

Value

return startup msg

Note

`getStartupMsg`

Examples

```
x <- getStartupMsg()
```

`getTableStyle` *getTableStyle*

Description

User defined selection that selects the style of table to display within the DEBrowser.

Usage

```
getTableStyle(dat = NULL, input = NULL, padj = c("padj"),
  foldChange = c("foldChange"), DEsection = TRUE)
```

Arguments

`dat,`	dataset
`input,`	input params
`padj,`	the name of the padj value column in the dataset
`foldChange,`	the name of the foldChange column in the dataset
`DEsection,`	if it is in DESection or not

Note

`getTableStyle`

Examples

```
x <- getTableStyle()
```

`getTextOnOff` *getTextOnOff*

Description

text on PCA plot on and off

Usage

```
getTextOnOff()
```

Note

`getTextOnOff`

Examples

```
    x <- getTextOnOff()
```

`getToolTipPCA` *getToolTipPCA*

Description

Prepares tooltiptext for PCA plot

Usage

```
getToolTipPCA(dat = NULL)
```

Arguments

`dat,`	data

Value

tooltip text

Examples

```
x <- getToolTipPCA()
```

`getToolTipText` *getToolTipText*

Description

Prepares tooltiptext for the second scatter plot in the plots page

Usage

```
getToolTipText(dat = NULL)
```

Arguments

`dat,`	data need to have following columns; padj, average, cond1 and cond2 values, log10padj, foldChange

Value

tooltip text

Examples

```
x <- getToolTipText()
```

`getUp` *getUp get up regulated data*

Description

getUp get up regulated data

Usage

```
getUp(filt_data = NULL)
```

Arguments

`filt_data,`	filt_data

Value

data

Examples

```
    x <- getUp()
```

`getUpDown` *getUpDown get up+down regulated data*

Description

getUpDown get up+down regulated data

Usage

```
getUpDown(filt_data = NULL)
```

Arguments

`filt_data,`	filt_data

Value

data

Examples

```
x <- getUpDown()
```

`get_state_id` *get_state_id*

Description

Helper to copy the bookmark to a user named directory

Usage

```
get_state_id(prev_url = NULL)
```

Arguments

`prev_url,`	prev url

Examples

```
x <- get_state_id()
```

`hideObj` *hideObj*

Description

Hides a shiny object.

Usage

```
hideObj(btns = NULL)
```

Arguments

`btns,`	hide group of objects with shinyjs

Examples

```
x <- hideObj()
```

`installpack` *installpack*

Description

install packages if they don’t exist display.

Usage

```
installpack(package_name = NULL, github = FALSE)
```

Arguments

`package_name,`	package name to be installed
`github,`	if github = true

Note

`installpack`

Examples

```
x <- installpack()
```

`link_brush` *link_brush*

Description

Modified linked brush object. A link brush function modified to be able to create non-reactive linked brush object for ggvis plots

Usage

```
link_brush()
```

Value

A list with components:

`input`	A function that takes a visualisation as an argument and adds an input brush to that plot
`selected`	A reactive providing a logical vector that describes which points are under the brush

Note

`link_brush` is very new and is likely to change substantially

Examples

```
lb <- link_brush()
```

`loadpack` *loadpack*

Description

load packages

Usage

```
loadpack(package_name = NULL)
```

Arguments

`package_name,`	package name to be loaded

Note

`loadpack`

Examples

```
x <- loadpack()
```

`loadpacks` *loadpacks*

Description

load initial packages

Usage

```
loadpacks()
```

Note

`loadpack`

Examples

```
x <- loadpacks()
```

`load_data` *load_data.*

Description

Loads user selected data to be used for DESeq

Usage

```
load_data(input = NULL, session = NULL)
```

Arguments

`input,`	input values
`session,`	if data is going to be loaded from json

Value

data

Examples

```
x<-load_data ()
```

`logSliderJScode` *logSliderJScode*

Description

Generates the log based slider to be used by the user within DEBrowser.

Usage

```
logSliderJScode(slidername = NULL)
```

Arguments

`slidername,`	id of the slider

Value

returns the slider values in log10 scale

Note

`logSliderJScode`

Examples

```
x <- logSliderJScode()
```

`mainScatter` *mainScatter*

Description

Creates the main scatter plot to be displayed within the main panel.

Usage

```
mainScatter(dat = NULL, lb = NULL, data_tooltip = NULL, x = NULL,
  y = NULL, domains = NULL, colors = NULL)
```

Arguments

`dat,`	dataframe that has log2FoldChange and log10padj values
`lb,`	the linked brush
`data_tooltip,`	toolstip specific to this plot
`x,`	the name of the x coordinate
`y,`	the name of the y coordinate
`domains,`	the domains to be colored
`colors,`	colors for each domain

Value

volcano plot

Examples

```
x <- mainScatter()
```

`MAPlot` *MAPlot*

Description

Prepares MA plot to be used within the main plot panel.

Usage

```
MAPlot(dat = NULL, lb = NULL, data_tooltip = NULL, domains = NULL,
  colors = NULL)
```

Arguments

`dat,`	dataframe that has log2FoldChange and log10padj values
`lb,`	the linked brush
`data_tooltip,`	toolstip specific to this plot
`domains,`	the domains to be colored
`colors,`	colors for each domain

Value

MA plot

Examples

```
x <- MAPlot()
```

`MAZoom` *MAZoom*

Description

Prepares the zoomed in version of the MA plot to be used within the main panel.

Usage

```
MAZoom(dat = NULL, data_tooltip = NULL, domains = NULL, colors = NULL)
```

Arguments

`dat,`	dataframe that has log2FoldChange and log10padj values
`data_tooltip,`	toolstip specific to this plot
`domains,`	the domains to be colored
`colors,`	colors for each domain

Value

zoomed MA plot

Examples

```
    x <- MAZoom()
```

`panel.cor` *panel.cor*

Description

Prepares the correlations for the all2all plot.

Usage

```
panel.cor(x, y, prefix = "rho=", cex.cor = 2, ...)
```

Arguments

`x,`	numeric vector x
`y,`	numeric vector y
`prefix,`	prefix for the text
`cex.cor,`	correlation font size
`...,`	additional parameters

Value

all2all correlation plots

Examples

```
panel.cor(c(1,2,3), c(4,5,6))
```

`panel.hist` *panel.hist*

Description

Prepares the historgram for the all2all plot.

Usage

```
panel.hist(x, ...)
```

Arguments

`x,`	a vector of values for which the histogram is desired
`...,`	any additional params

Value

all2all histogram plots

Examples

```
panel.hist(1)
```

`plot_pca` *plot_pca*

Description

Plots the PCA results for the selected dataset.

Usage

```
plot_pca(dat = NULL, pcx = 1, pcy = 2, metadata = NULL, color = NULL,
  shape = NULL, size = NULL, textonoff = "Off", legendSelect = "fill")
```

Arguments

`dat,`	data
`pcx,`	x axis label
`pcy,`	y axis label
`metadata,`	additional data
`color,`	color for plot
`shape,`	shape for plot
`size,`	size of the plot
`textonoff,`	text on off
`legendSelect,`	select legend

Value

pca list

Examples

```
load(system.file("extdata", "demo", "demodata.Rda",
        package="debrowser"))
metadata<-cbind(colnames(demodata[,2:7]),
        colnames(demodata[,2:7]),
        c(rep("Cond1",3), rep("Cond2",3)))
colnames(metadata)<-c("samples", "color", "shape")

a <- plot_pca(getNormalizedMatrix(
        demodata[rowSums(demodata[,2:7])>10,2:7]),
        metadata = metadata, color = "samples",
        size = 5, shape = "shape")
```

`prepAddQCPlots` *prepAddQCPlots*

Description

prepares IQR and density plots

Usage

```
prepAddQCPlots(data = NULL, input = NULL)
```

Arguments

`data,`	barplot data
`input,`	user input params

Examples

```
prepAddQCPlots()
```

`prepDataContainer` *prepDataContainer*

Description

Prepares the data container that stores values used within DESeq.

Usage

```
prepDataContainer(data = NULL, counter = NULL, input = NULL)
```

Arguments

`data,`	loaded dataset
`counter,`	the number of comparisons
`input,`	input parameters

Value

data

Examples

```
x <- prepDataContainer()
```

`prepDataForQC` *prepDataForQC*

Description

Prepares selected data for QC plots.

Usage

```
prepDataForQC(dataset = NULL, input = NULL)
```

Arguments

`dataset,`	loaded dataset
`input,`	input

Value

data

Examples

```
x <- prepDataForQC()
```

`prepDEOutput` *prepDEOutput*

Description

Prepares the output data from DE analysis to be used within DEBrowser

Usage

```
prepDEOutput(data = NULL, cols = NULL, conds = NULL, inputconds = NULL,
  i = NULL, input = NULL)
```

Arguments

`data,`	loaded dataset
`cols,`	columns
`conds,`	conds
`inputconds,`	inputconds
`i,`	selected comparison number
`input,`	input

Value

data

Examples

```
    x <- prepDEOutput()
```

`push` *push*

Description

Push an object to the list.

Usage

```
push(l, ...)
```

Arguments

`l,`	that are going to push to the list
`...,`	list object

Value

combined list

Examples

```
mylist <- list()
newlist <- push ( 1, mylist )
```

`readMetaData` *readMetaData*

Description

read metadata file

Usage

```
readMetaData(input = NULL)
```

Arguments

`input,`	input values

Note

`readMetaData`

Examples

```
x <- readMetaData()
```

`removeBookmark` *removeBookmark*

Description

remove saved state

Usage

```
removeBookmark(ID = NULL, username = NULL)
```

Arguments

`ID,`	prev url
`username,`	username

Examples

```
x <- removeBookmark()
```

`removeCols` *removeCols*

Description

remove unnecessary columns

Usage

```
removeCols(cols = NULL, dat = NULL)
```

Arguments

`cols,`	columns that are going to be removed from data frame
`dat,`	data

Value

data

Examples

```
x <- removeCols()
```

`round_vals` *round_vals*

Description

Plot PCA results.

Usage

```
round_vals(l)
```

Arguments

`l,`	the value

Value

round value

Examples

```
x<-round_vals(5.1323223)
```

`runBayseq` *runBayseq*

Description

Run Bayseq algorithm on the selected conditions. Output is to be used for the interactive display.

Usage

```
runBayseq(data = NULL, columns = NULL, conds = NULL, rowsum.filter = 10)
```

Arguments

`data,`	A matrix that includes all the expression raw counts, rownames has to be the gene, isoform or region names/IDs
`columns,`	is a vector that includes the columns that are going to be analyzed. These columns has to match with the given data.
`conds,`	experimental conditions. The order has to match with the column order
`rowsum.filter,`	regions/genes/isoforms with total count (across all samples) below this value will be filtered out

Value

BaySeq results

Examples

```
x <- runBayseq()
```

`runDE` *runDE*

Description

Run DE algorithms on the selected parameters. Output is to be used for the interactive display.

Usage

```
runDE(data = NULL, columns = NULL, conds = NULL, pars = NULL)
```

Arguments

`data,`	A matrix that includes all the expression raw counts, rownames has to be the gene, isoform or region names/IDs
`columns,`	is a vector that includes the columns that are going to be analyzed. These columns has to match with the given data.
`conds,`	experimental conditions. The order has to match with the column order
`pars,`	all params for the de methods

Value

de results

Examples

```
x <- runDE()
```

`runDESeq` *runDESeq*

Description

Run DESeq2 algorithm on the selected conditions. Output is to be used for the interactive display.

Usage

```
runDESeq(data, columns, conds, fitType = c("parametric", "local", "mean"),
  non_expressed_cutoff = 10)
```

Arguments

`data,`	A matrix that includes all the expression raw counts, rownames has to be the gene, isoform or region names/IDs
`columns,`	is a vector that includes the columns that are going to be analyzed. These columns has to match with the given data.
`conds,`	experimental conditions. The order has to match with the column order
`fitType,`	DESeq2 fitType, it can be 'parametric', 'local', 'mean'.
`non_expressed_cutoff,`	to remove unexpressed regions/genes/isoforms this cutoff is used

Value

deseq2 results

Examples

```
x <- runDESeq(data<-NULL, columns<-c())
```

`runDESeq2` *runDESeq2*

Description

Run DESeq2 algorithm on the selected conditions. Output is to be used for the interactive display.

Usage

```
runDESeq2(data = NULL, columns = NULL, conds = NULL,
  fitType = c("parametric", "local", "mean"), betaPrior = 0,
  testType = c("Wald", "LRT"), rowsum.filter = 10)
```

Arguments

`data,`	A matrix that includes all the expression raw counts, rownames has to be the gene, isoform or region names/IDs
`columns,`	is a vector that includes the columns that are going to be analyzed. These columns has to match with the given data.
`conds,`	experimental conditions. The order has to match with the column order
`fitType,`	either "parametric", "local", or "mean" for the type of fitting of dispersions to the mean intensity. See estimateDispersions for description.
`betaPrior,`	whether or not to put a zero-mean normal prior on the non-intercept coefficients See nbinomWaldTest for description of the calculation of the beta prior. By default, the beta prior is used only for the Wald test, but can also be specified for the likelihood ratio test.
`testType,`	either "Wald" or "LRT", which will then use either Wald significance tests (defined by nbinomWaldTest), or the likelihood ratio test on the difference in deviance between a full and reduced model formula (defined by nbinomLRT)
`rowsum.filter,`	regions/genes/isoforms with total count (across all samples) below this value will be filtered out

Value

deseq2 results

Examples

```
x <- runDESeq2()
```

---

runEdgeR *runEdgeR*

Description

Run EdgeR algorithm on the selected conditions. Output is to be used for the interactive display.

Usage

```
runEdgeR(data = NULL, columns = NULL, conds = NULL, normfact = c("TMM",
  "RLE", "upperquartile", "none"), dispersion = 0, testType = c("glmLRT",
  "exactTest"), rowsum.filter = 10)
```

Arguments

`data,`	A matrix that includes all the expression raw counts, rownames has to be the gene, isoform or region names/IDs
`columns,`	is a vector that includes the columns that are going to be analyzed. These columns has to match with the given data.
`conds,`	experimental conditions. The order has to match with the column order
`normfact,`	Calculate normalization factors to scale the raw library sizes. Values can be "TMM","RLE","upperquartile","none".
`dispersion,`	either a numeric vector of dispersions or a character string indicating that dispersions should be taken from the data object. If a numeric vector, then can be either of length one or of length equal to the number of genes. Allowable character values are "common", "trended", "tagwise" or "auto". Default behavior ("auto" is to use most complex dispersions found in data object.
`testType,`	exactTest or glmLRT. exactTest: Computes p-values for differential abundance for each gene between two digital libraries, conditioning on the total count for each gene. The counts in each group as a proportion of the whole are assumed to follow a binomial distribution. glmLRT: Fit a negative binomial generalized log-linear model to the read counts for each gene. Conduct genewise statistical tests for a given coefficient or coefficient contrast.
`rowsum.filter,`	regions/genes/isoforms with total count (across all samples) below this value will be filtered out

Value

edgeR results

Examples

```
x <- runEdgeR()
```

`runHeatmap` *runHeatmap*

Description

Creates a heatmap based on the user selected parameters within shiny.

Usage

```
runHeatmap(data, title = "Title", dend = "both", names = FALSE,
  clustering_method = c("ward.D2", "complete", "single", "average",
  "mcquitty", "median", "centroid"), distance_method = c("euclidean", "cor",
  "maximum", "manhattan", "canberra", "binary", "minkowski"),
  interactive = FALSE)
```

Arguments

`data,`	a matrixthat includes expression values
`title,`	title of the heatmap
`dend,`	dendogram
`names,`	a flag to show the rownames
`clustering_method`	= c('complete', 'ward.D2', 'single', 'average', 'mcquitty', 'median' , 'centroid')
`distance_method`	= c('cor','euclidean', 'maximum', 'manhattan', 'canberra', 'binary' ,'minkowski')
`interactive,`	interactive heatmap

Value

heatmap.2 plot

Examples

```
x <- runHeatmap(mtcars)
```

`runLimma` *runLimma*

Description

Run Limma algorithm on the selected conditions. Output is to be used for the interactive display.

Usage

```
runLimma(data = NULL, columns = NULL, conds = NULL, normfact = c("none",
  "TMM", "RLE", "upperquartile"), fitType = c("ls", "robust"),
  normBet = c("none", "scale", "quantile", "cyclicloess", "Aquantile",
  "Gquantile", "Rquantile", "Tquantile"), rowsum.filter = 10)
```

Arguments

`data,`	A matrix that includes all the expression raw counts, rownames has to be the gene, isoform or region names/IDs
`columns,`	is a vector that includes the columns that are going to be analyzed. These columns has to match with the given data.
`conds,`	experimental conditions. The order has to match with the column order
`normfact,`	Calculate normalization factors to scale the raw library sizes. Values can be "TMM","RLE","upperquartile","none".
`fitType,`	fitting method; "ls" for least squares or "robust" for robust regression
`normBet,`	Normalizes expression intensities so that the intensities or log-ratios have similar distributions across a set of arrays.
`rowsum.filter,`	regions/genes/isoforms with total count (across all samples) below this value will be filtered out

Value

Limma results

Examples

```
x <- runLimma()
```

`run_pca` *run_pca*

Description

Runs PCA on the selected dataset.

Usage

```
run_pca(x = NULL, retx = TRUE, center = TRUE, scale = TRUE)
```

Arguments

`x,`	dataframe with experiment data
`retx,`	specifies if the data should be returned
`center,`	center the PCA (Boolean)
`scale,`	scale the PCA (Boolean)

Value

pca list

Examples

```
load(system.file("extdata", "demo", "demodata.Rda",
    package="debrowser"))
pca_data<-run_pca(getNormalizedMatrix(
    demodata[rowSums(demodata[,2:7])>10,2:7]))
```

`saveQCPlot` *saveQCPlot*

Description

Saves the current QC plot selection to the users local disk.

Usage

```
saveQCPlot(filename = NULL, input = NULL, datasetInput = NULL,
  cols = NULL, conds = NULL, inputQCPlot = NULL)
```

Arguments

`filename,`	filename
`input,`	input params
`datasetInput,`	dataset
`cols,`	selected columns
`conds,`	selected conditions
`inputQCPlot,`	clustering method and distance method

Note

`saveQCPlot`

Examples

```
saveQCPlot()
```

`scatterZoom` *scatterZoom*

Description

Displays the zoomed in version of the plot to be viewed within the main panel.

Usage

```
scatterZoom(dat = NULL, data_tooltip = NULL, x = NULL, y = NULL,
  domains = NULL, colors = NULL)
```

Arguments

`dat,`	dataframe that has log2FoldChange and log10padj values
`data_tooltip,`	toolstip specific to this plot
`x,`	the name of the x coordinate
`y,`	the name of the y coordinate
`domains,`	the domains to be colored
`colors,`	colors for each domain

Value

zoomed scatter plot

Examples

```
x <- scatterZoom()
```

`selectBatchEffect` *selectBatchEffect*

Description

Batch effect column selection

Usage

```
selectBatchEffect(input = NULL, selectname = "batchselect",
  label = "Batch effect correction column")
```

Arguments

`input,`	input values
`selectname,`	name of the select box
`label,`	label of the select box

Note

`selectBatchEffect`

Examples

```
x <- selectBatchEffect()
```

---

selectConditions *selectConditions*

Description

Selects user input conditions, multiple if present, to be used in DESeq.

Usage

```
selectConditions(Dataset = NULL, choicecounter, input = NULL,
  loadingJSON = NULL)
```

Arguments

`Dataset,`	used dataset
`choicecounter,`	total number of comparisons
`input,`	input params
`loadingJSON,`	loads from json

Value

the panel for go plots;

Note

`selectConditions`

Examples

```
x<- selectConditions()
```

`selectedInput` *selectedInput*

Description

Selects user input conditions to run in DESeq.

Usage

```
selectedInput(id = NULL, num = 0, default = NULL, input = NULL)
```

Arguments

`id,`	input id
`num,`	panel that is going to be shown
`default,`	default text
`input,`	input params

Examples

```
x <- selectedInput()
```

`setFilterParams` *setFilterParams*

Description

It sets the filter parameters

Usage

```
setFilterParams(session = NULL, input = NULL)
```

Arguments

`session,`	session variable
`input,`	input parameters

Examples

```
x <- setFilterParams()
```

`showObj` *showObj*

Description

Displays a shiny object.

Usage

```
showObj(btns = NULL)
```

Arguments

`btns,`	show group of objects with shinyjs

Examples

```
x <- showObj()
```

`startDEBrowser` *startDEBrowser*

Description

Starts the DEBrowser to be able to run interactively.

Usage

```
startDEBrowser()
```

Value

the app

Note

`startDEBrowser`

Examples

```
startDEBrowser()
```

`textareaInput` *textareaInput*

Description

Generates a text area input to be used for gene selection within the DEBrowser.

Usage

```
textareaInput(id, label, value, rows = 20, cols = 35,
  class = "form-control")
```

Arguments

`id,`	id of the control
`label,`	label of the control
`value,`	initial value
`rows,`	the # of rows
`cols,`	the # of cols
`class,`	css class

Examples

```
x <- textareaInput("genesetarea", "Gene Set",
    "Fgf21", rows = 5, cols = 35)
```

`togglePanels` *togglePanels*

Description

User defined toggle to display which panels are to be shown within DEBrowser.

Usage

```
togglePanels(num = NULL, nums = NULL, session = NULL)
```

Arguments

`num,`	selected panel
`nums,`	all panels
`session,`	session info

Note

`togglePanels`

Examples

```
x <- togglePanels()
```

`volcanoPlot` *volcanoPlot*

Description

Prepares volcano plot to be used within the DEBrowser.

Usage

```
volcanoPlot(dat = NULL, lb = NULL, data_tooltip = NULL, domains = NULL,
  colors = NULL)
```

Arguments

`dat,`	dataframe that has log2FoldChange and log10padj values
`lb,`	the linked brush
`data_tooltip,`	toolstip specific to this plot
`domains,`	the domains to be colored
`colors,`	colors for each domain

Value

volcano plot

Examples

```
x <- volcanoPlot()
```

`volcanoZoom` *volcanoZoom*

Description

Prepares the zoomed in version of the volcano plot to be used within the Debrowser.

Usage

```
volcanoZoom(dat = NULL, data_tooltip = NULL, domains = NULL,
  colors = NULL)
```

Arguments

`dat,`	dataframe that has log2FoldChange and log10padj values
`data_tooltip,`	toolstip specific to this plot
`domains,`	the domains to be colored
`colors,`	colors for each domain

Value

zoomed volcano plot

Examples

```
x <- volcanoZoom()
```

Index