Package 'RUtil'

April 7, 2011

Version 0.03

License GPL (>= 2)

Description A package collecting useful R code snippets from different projects made by Raivo Kolde.

Title Raivos util functions

Author Raivo Kolde <rkolde@gmail.com>

Maintainer Raivo Kolde <rkolde@gmail.com>

Depends R (>= 2.10)

Imports plyr, reshape, stringr, ggplot2, RCurl, grid

Collate 'bubbleplot.r' 'ggplot2_themes.r' 'gprofiler.r' 'make_unique_comb.r' 'ROC.r' 'RUtil-package.r'

R topics documented:

`RUtil-package` *A set of useful functions by Raivo...*

Description

A set of useful functions by Raivo

bubbleplot *Function to draw the bubbleplots - an alternative to Venn diagram...*

Description

Function to draw the bubbleplots - an alternative to Venn diagram

Usage

```
bubbleplot(lists, percentage=TRUE)
```

Arguments

lists	input element lists
percentage	logical showing if percentages or raw numbers are displayed

Details

Function to draw the bubbleplots - an alternative to Venn diagram

Value

Returns nothing only draws the picture

Author(s)

Raivo Kolde <rkolde@gmail.com>

Examples

```
bubbleplot(list(Asadsa = sample(letters, 14), Gadsa = sample(letters, 9)))
bubbleplot(list(Badsad = sample(letters, 16), Asadsa = sample(letters, 14), Gadsa = sampl
```

calcAUC *Calculate AUC score based on the given curves...*

Description

Calculate AUC score based on the given curves

Usage

```
calcAUC(curves, bootstrap=0)
```

Arguments

curves	curves as calculated by calcCurve
bootstrap	number of bootstrap samples for finding confidence intervals

Details

Calculate AUC score based on the given curves

Value

returns a data frame with auc values for each class, optionally also with bootstrapped confidence intervals

Author(s)

Raivo Kolde <rkolde@gmail.com>

`calcCurve` *Calculate ROC curves based on ordered class data...*

Description

Calculate ROC curves based on ordered class data

Usage

```
calcCurve(data)
```

Arguments

`data`	Data as described in `data`

Details

Calculate ROC curves based on ordered class data

Value

Returns a data frame like data, but with additional columns TP and TN

Author(s)

Raivo Kolde <rkolde@gmail.com>

`gconvert` *Convert gene ID-s.*

Description

Convert gene ID-s.

Usage

```
gconvert(ids, organism="hsapiens", target="ENSG", df=T)
```

Arguments

`ids`	gene list.
`organism`	gene list origin.
`target`	target ID.
`df`	logical showing if the output will be a data.frame or list.

Details

Wrapper for g:Convert web toolkit.

Value

The output can be either list or data.frame. List has an entry for every input gene. Data frame is just a two column table with inputs and corrsponding outputs. The input names may be duplicated.

Author(s)

Juri Reimand <jyri.reimand@ut.ee>, Raivo Kolde <rkolde@gmail.com>

References

J. Reimand, M. Kull, H. Peterson, J. Hansen, J. Vilo: g:Profiler - a web-based toolset for functional profiling of gene lists from large-scale experiments (2007) NAR 35 W193-W200

Examples

```
gconvert(c("Klf4", "Pax5", "Sox2", "Nanog"), organism = "mmusculus")
```

`generate_gcocoa_query` *Generate query for g:Cocoa web tool...*

Description

Generate query for g:Cocoa web tool

Usage

```
generate_gcocoa_query(glist)
```

Arguments

`glist`	a list of vectors of gene names

Details

Function that prints out a g:Gogoa query string given a list of gene name vectors.

Value

Prints out the query string

Author(s)

Raivo Kolde <rkolde@gmail.com>

Examples

```
glist = list(a = c("pax6", "klf9"), b = c("nanog", "Pou5f1"))
generate_gcocoa_query(glist)
```

`gorth` *Find orthologs.*

Description

Find orthologs.

Usage

```
gorth(genelist, source_organism="mmusculus", target_organism="hsapiens", mthresh
```

Arguments

genelist	list of gene names to be translated.
source_organism	organism of the input genes.
target_organism	name for the target organism.
mthreshold	how many ortholog names per gene to show
df	logical showing if the output will be a data.frame or list.

Details

Wrapper for g:Orth web toolkit. The organism names are constructed, by combining the first letter of the name and family name. For example human - 'hsapiens' and mouse- 'mmusculus'

To alleviate the problem of having many orthologs per gene (most of them uninformative) one can set a threshold of how many results to show. The programs tries to find the most informative by selecting the most popular ones.

Value

The output can be either list or data.frame. List has an entry for every input gene. Data frame is just a two column table with inputs and corrsponding outputs. The input names may be duplicated.

Author(s)

Raivo Kolde <rkolde@gmail.com>, Juri Reimand <jyri.reimand@ut.ee>

References

J. Reimand, M. Kull, H. Peterson, J. Hansen, J. Vilo: g:Profiler – a web-based toolset for functional profiling of gene lists from large-scale experiments (2007) NAR 35 W193-W200

Examples

```
gorth(c("Klf4", "Pax5", "Sox2", "Nanog"), source_organism = "mmusculus", target_organism
```

---

gprofiler *Annotate gene list functionally.*

Description

Annotate gene list functionally.

Usage

```
gprofiler(organism="scerevisiae", query, ordered_query=0, significant=1)
```

Arguments

organism	gene list origin
query	vector of gene names or list of such vectors
ordered_query	when output gene lists are ranked one can use this option to get GSEA style p-values.
significant	logical indicating if all or only statistically significant results should be returned.

Details

Wrapper for g:Profiler web toolkit for finding enrichments in gene lists.

Value

Data frame with the Enricment analysis results. If input consisted of several lists the corresponding list is indicated with a variable 'query number'

Author(s)

Juri Reimand <jyri.reimand@ut.ee>

References

J. Reimand, M. Kull, H. Peterson, J. Hansen, J. Vilo: g:Profiler - a web-based toolset for functional profiling of gene lists from large-scale experiments (2007) NAR 35 W193-W200

Examples

```
gprofiler(c("Klf4", "Pax5", "Sox2", "Nanog"), organism = "mmusculus")
```

---

make_unique_comb *Function for producing all unique combinations of elements from one vector.*

Description

Function for producing all unique combinations of elements from one vector.

Usage

```
make_unique_comb(l)
```

Arguments

l	vector of some values

Details

The function creates all unique combinations of different values in vector l.

Value

A matrix with two columns. Rownames are the element names pasted togeteher and separated by "_". The rows are ordered alphabetically according to rownames. The two values in each row are also ordered alphabetically.

Author(s)

Raivo Kolde <rkolde@gmail.com>

Examples

```
make_unique_comb(letters[1:3])
```

---

plotROC *Plot ROC curve...*

Description

Plot ROC curve

Usage

```
plotROC(data, bootstrap=100, colours=NA, text_size=4, all_replicates=TRUE)
```

Arguments

`data`	input data.frame
`colours`	what colours should the lines be
`text_size`	AUC text size
`bootstrap`	number of bootstrap samples
`all_replicates`	logical if we show the repilcates as well

Details

Input data frame has to contain two columns Probability and Class. Probability refers to some score accordind to which the data is ordered and Class referes to if the observation was positive or negative. If several lines are needed, additional column algorithm should be provided. The ROC curves and AUC-s are calculated by functions `calcCurve` and `calcAUC`. By default the AUC values get also confidence intervals, which are calculated using bootstrap. The number of bootstrap samples taken can be set with parameter bootstrap. If desired one can also enter replicate experiments to calculate confidence intervals. For that the dataset has to contain also Column Replicate.

Value

Decription of return value

Author(s)

Raivo Kolde <rkolde@gmail.com>

Examples

```
data = data.frame(Probability = runif(300))
data$Class = rbinom(300, 1, prob = 1 - data$Probability)
#plotROC(data)

data$Algorithm = sample(c("A", "B", "C"), 300, replace = TRUE)
#plotROC(data)

data$Replicate = factor(sample(1:5, nrow(data), replace = TRUE))
#plotROC(data)

data$Replicate = factor(sample(1:5, nrow(data), replace = TRUE))
#plotROC(data)
```

---

theme_bw_raivo *Theme_bw without ugly whitespace on x axis...*

Description

Theme_bw without ugly whitespace on x axis

Usage

```
theme_bw_raivo(base_size=12, base_family="")
```

Arguments

base_size	base font size
base_family	base font family

Details

Theme_bw without ugly whitespace on x axis

Author(s)

Raivo Kolde <rkolde@gmail.com>

Index