Package: muRtools (via r-universe)

August 26, 2024

Title Mueller's R tools

Description Fabian's custom plotting functions and utilities

Author Fabian Mueller

Maintainer Fabian Mueller <fmueller@mpi-inf.mpg.de>

Date 2023-06-20

Suggests gplots, plotrix, BSgenome.Hsapiens.UCSC.hg19, BSgenome.Hsapiens.NCBI.GRCh38, BSgenome.Mmusculus.UCSC.mm9, BSgenome.Mmusculus.UCSC.mm10

Imports muLogR, ggplot2, data.table, GenomicRanges, GenomeInfoDb

License GPL-2

Version 0.9.6

Encoding UTF-8

Collate 'analysisConfig.R' 'clustering.R' 'colors.R' 'commandLine.R' 'develTools.R' 'dimRed.R' 'encodeProjectTools.R' 'enrichmentTools.R' 'epigeneticsConfig.R' 'epigeneticsData.R' 'epigeneticsUtils.R' 'genomicRegions.R' 'heatmaps.R' 'listOps.R' 'lolaUtils.R' 'miscPlots.R' 'muKDEplots.R' 'muMakeTrans.R' 'muPairs.R' 'muRtools.R' 'muScatter.R' 'stringOps.R' 'mu_ggplots2_ext.R' 'normalize.R' 'rnbeads.R' 'statTest.R' 'txdb.R' 'wrappers.R'

RoxygenNote 7.2.3

Repository https://blaserlab.r-universe.dev

RemoteUrl https://github.com/demuellae/muRtools

RemoteRef HEAD

RemoteSha f19bda1707512fe8d3b97df0c22d4934e62f6aaa

Contents

`aggregateDf` *aggregateDf*

Description

Wrapper around `aggregate` to merge rows of a data frame based on a specified grouping. Merging is done by either returning the unique value for each group or if multiple different values exists converting them to character and concatenating by ";"

Usage

```
aggregateDf(df, groupBy)
```

Arguments

`df`	`data.frame` to aggregate
`groupBy`	vector whose unique values indicates the grouping of the rows in the data frame

Value

aggregated `data.frame`

Author(s)

Fabian Mueller

`bed2GRanges` *bed2GRanges*

Description

loads a bed file and converts it to a GRanges object. Note that bed files are 0-based, right-exclusive by definition The output of this function will be 1-based, right inclusive as defined by GRanges.

Usage

```
bed2GRanges(fname, assembly = NA)
```

Arguments

`assembly`	genome assembly
`id`	region set id

Value

a GRanges object containing the region set

Examples

```
reg.fname <- system.file(file.path("extdata","deep_chip_ctrl_regions.hg19.bed"), package = "muRtools")
regs <- bed2GRanges(reg.fname)
```

---

bedTobigBed *bedTobigBed*

Description

Convert a bed file to bigBed. requires the 'bedToBigBed' tool

Usage

```
bedTobigBed(
  bedFn,
  chromSizes,
  bbFn = paste0(gsub("\\.bed$", "", bedFn), ".bb"),
  bedToBigBed = "bedToBigBed"
)
```

Arguments

bedFn	filename of the bed file
chromSizes	named vector of chromosome sizes
bbFn	filename to save the bigBed file to
bedToBigBed	executable of the 'bedToBigBed' tool

Value

nothing of particular interest

camel2underscore *camel2underscore*

Description

converts camel case to underscores in a (vector of) string(s)

Usage

```
camel2underscore(x)
```

Arguments

a	string or string vector

Value

the converted string(s)

Examples

```
underscore2camel("bla_blubb")
```

chordDiagramFromContingencyTable
chordDiagramFromContingencyTable

Description

Plot a bipartite chord diagram (circular Sankey diagram with 2 categories) for a contingency matrix the values of two matrices

Usage

```
chordDiagramFromContingencyTable(
  contTab,
  chordColorByCol = FALSE,
  cs_rows = colpal.mu.cat,
  cs_columns = colpal.mu.cat,
  ...
)
```

Arguments

contTab	matrix or table containing the contingency matrix
chordColorByCol	color chords by column instead of by row
cs_rows	color scheme to use for the rows of the matrix
cs_columns	color scheme to use for the rows of the matrix
...	passed on to chordDiagram

Value

nothing of particular interest (include this function while plotting).

Author(s)

Fabian Mueller

Examples

```
# contingency table of air quality quantile by month
contTab <- with(airquality, table(cut(Temp, quantile(Temp)), Month))
names(dimnames(contTab))[1] <- "quantile"
chordDiagramFromContingencyTable(contTab)
```

---

col.text.2.hex *col.text.2.hex*

Description

convert a color from a text string to a hex string (only works with scalars, not on vectors)

Usage

```
col.text.2.hex(ss, alpha = 255)
```

Arguments

ss	color string to be transformed
alpha	maximum alpha

Value

a string containing the hex code of the color

Examples

```
col.text.2.hex("dark blue")
```

colorize.value *colorize.value*

Description

transforms a value to a color based on a colorscale and a value scale

Usage

```
colorize.value(
  val,
  rng = c(-1, 1),
 colscheme.col.val = c(gplots::colorpanel(100, "blue", "white"), gplots::colorpanel(100,
    "white", "red"))
)
```

Arguments

val	value to be transformed
rng	value range
colscheme.col.val	Color palette to pick a color from

Value

color of the value

Examples

```
colorize.value(0.5,rng=c(-1,1), colscheme.col.val=c(colorpanel(100,"blue","white"),colorpanel(100,"white","red"
```

colpal.cb *Custom Color Paletes*

Description

colpal.cb color blind friendly color palettes (adapted from http://wiki.stdout.org/rcookbook/Graphs/Colors

Usage

```
colpal.cb

colpal.bde

colpal.nature

colpal.mu.cat

colpal.iwh.cb01

colpal.solarextra

colpals.topo

colpal.PhFr.a

colpal.miniblaze

colpal.corpid

colpals.games
```

Format

An object of class `character` of length 9.

An object of class `character` of length 13.

An object of class `character` of length 17.

An object of class `character` of length 18.

An object of class `character` of length 23.

An object of class `character` of length 9.

An object of class `list` of length 2.

An object of class `character` of length 9.

An object of class `character` of length 6.

An object of class `character` of length 19.

An object of class `list` of length 6.

Examples

```
plotColpal(colpal.cb)
plotColpal(colpal.bde)
plotColpal(colpal.nature)
plotColpal(colpal.mu.cat)
plotColpal(colpal.iwh.cb01)
plotColpal(colpal.solarextra)
plotColpal(colpals.topo[["dem_cut"]])
```

```
plotColpal(colpal.PhFr.a)
plotColpal(colpal.PhFr.a)
plotColpal(colpal.corpid)
plotColpal(colpals.games[["rollgalaxy"]])
```

`colpal.cont` *colpal.cont*

Description

Get a continuous color palette

Usage

```
colpal.cont(n = 3, name = "viridis", ...)
```

Arguments

`n`	number of colors returned
`name`	name of the color palette
`...`	arguments passed to other functions

Value

a character vector containing n colors

Author(s)

Fabian Mueller

Examples

```
plotColpal(colpal.cont(5, "viridis"))
plotColpal(colpal.cont(5, "cb.BrBG"))
plotColpal(colpal.cont(9, "solarextra"))
plotColpal(colpal.cont(9, "cptcity.schwarzwald_cont"))
plotColpal(colpal.cont(9, "cptcity.europe_7"))
plotColpal(colpal.cont(9, "cptcity.spain"))
plotColpal(colpal.cont(9, "cptcity.nordisk"))
plotColpal(colpal.cont(9, "cptcity.cmocean_delta"))
plotColpal(colpal.cont(9, "cptcity.colombia"))
plotColpal(colpal.cont(9, "cptcity.blue_tan_d14"))
plotColpal(colpal.cont(9, "cptcity.arendal_temperature"))
plotColpal(colpal.cont(9, "cptcity.jjg_misc_temperature"))
plotColpal(colpal.cont(9, "cptcity.jjg_neo10_elem_rain"))
plotColpal(colpal.cont(9, "cptcity.es_vintage_57"))
plotColpal(colpal.cont(9, "cptcity.es_skywalker_02"))
```

colpal.histone *Custom Color Paletes for epigenetic modifications*

Description

Named vectors of colors for different modifications

`colpal.histone` Histone modifications

Usage

```
colpal.histone

colpal.histone.ihec

colgrad.methylation.rb

colgrad.methylation.yb
```

Format

An object of class `character` of length 10.

An object of class `character` of length 8.

An object of class `character` of length 3.

An object of class `character` of length 3.

Examples

```
library(gplots)
pie(rep(1,length(colpal.histone)), labels=names(colpal.histone), col=colpal.histone)
library(gplots)
pie(rep(1,length(colpal.histone.ihec)), labels=names(colpal.histone.ihec), col=colpal.histone.ihec)
library(plotrix)
cp <- colorpanel(100,colgrad.methylation.rb["low"],colgrad.methylation.rb["mid"],colgrad.methylation.rb["high"]
plot.new()
gradient.rect(0,0,1,1,col=cp,nslices=length(cp),gradient="x",border=NA)
library(plotrix)
cp <- colorpanel(100,colgrad.methylation.yb["low"],colgrad.methylation.yb["mid"],colgrad.methylation.yb["high"]
plot.new()
gradient.rect(0,0,1,1,col=cp,nslices=length(cp),gradient="x",border=NA)
```

combinationList *combinationList*

Description

get a list of all combinations of vectors. Basically a wrapper around `expand.grid`

Usage

```
combinationList(...)
```

Arguments

`...`	vectors of elements. Ideally named

Value

a list containing all combinations of elements in the input. Each element contains a unique combination

Examples

```
combinationList(a=letters[1:5], A=LETTERS[1:3], i=1:4)
```

containsHistoneModStr *Methods for recognizing histone modifications from strings*

Description

Methods for recognizing histone modifications from strings

Usage

```
containsHistoneModStr(s)

getHistoneFromHistoneModStr(s)

getAaTypeFromHistoneModStr(s)

getAaPosFromHistoneModStr(s)

getModFromHistoneModStr(s)

normalizeHistoneModStr(s)

matchHistoneModStr(s1, s2)
```

Arguments

s	a string
s1	a string
s2	a string
	containsHistoneModStr Does a string contain the pattern for histone modifications?
	getHistoneFromHistoneModStr Retrive the histone from a string containing a histone modification pattern
	getAaTypeFromHistoneModStr Retrive the amino acid type from a string containing a histone modification pattern
	getAaPosFromHistoneModStr Retrive the amino acid position from a string containing a histone modification pattern
	getModFromHistoneModStr Retrive the modification from a string containing a histone modification pattern
	normalizeHistoneModStr normalize the histone modification if contained in a string
	matchHistoneModStr Do two strings contain the same histone modifications

Examples

```
s1 <- "H3K4me3"
s2 <- "h3k04ME3"
s3 <- "blubb5A27me3"
s4 <- "h3k27ac"
containsHistoneModStr(s1)
containsHistoneModStr(s3)
containsHistoneModStr(s4)
getHistoneFromHistoneModStr(s1)
getAaTypeFromHistoneModStr(s1)
getAaPosFromHistoneModStr(s1)
getModFromHistoneModStr(s1)
normalizeHistoneModStr(s1)
normalizeHistoneModStr(c(s1,s2,s3,s4))
matchHistoneModStr(s1,s2)
matchHistoneModStr(s1,s4)
```

---

countPairwiseOverlaps *countPairwiseOverlaps*

Description

Fast counting of pairwise overlaps between two lists of region sets

Usage

```
countPairwiseOverlaps(grl1, grl2, ...)
```

Arguments

grl1	list of GRanges or GRangesList object 1
grl2	list of GRanges or GRangesList object 2
...	arguments passed on to findOverlaps

Value

an integer matrix containing pairwise overlaps between elements in grl1 and grl1

---

create.densityScatter *create.densityScatter*

Description

Creates a density scatterplot highlighting points in sparsely populated plot regions as well as points marked as special in a seperate color

Usage

```
create.densityScatter(
  df2p,
  is.special = NULL,
  dens.subsample = FALSE,
  dens.special = TRUE,
  sparse.points = 0.01,
  dens.n = 100,
  add.text.cor = FALSE
)
```

Arguments

df2p	data.frame to be plotted. Only the fist two columns are taken into account as x and y coordinates respectively
is.special	boolean vector of length equal to the number of rows in df2p. Specifies which points should be highlighed seperately in a different color
dens.subsample	if the number of points exceeds this number, subsample the number of points for the density estimation to that number. Any non-numeric value disables subsampling.
dens.special	Flag indicating whether the points of the special population should be colored according to their density
sparse.points	Either percentage (<=1,>=0) or the absolute number of points in the sparsely populated area that should be drawn seperately. A value of 0 means that these points will not be drawn.
dens.n	passed on to ggplot2::stat_density2d: argument: n
add.text.cor	flag indicating whether a text token with the correlation coefficient should be included in the lower right corner of the plot

Value

ggplot object

Author(s)

Fabian Mueller (RnBeads)

Examples

```
d <- data.frame(x=rnorm(1000),y=rnorm(1000))
s <- rep(FALSE,1000)
s[sample(1:length(s),100)] <- TRUE
create.densityScatter(d,s)
```

---

densRanks *densRanks*

Description

Rank the points accordind to density of the region they fall in. Densities are computed as Kernel Density estimates. The method and parameters are implemented in analogy to grDevices::densCols

Usage

```
densRanks(x, y = NULL, nbin = 128, bandwidth)
```

Arguments

x	x-coordinate
y	y-coordinate
nbin	number of bins
bandwidth	bandwidth

Author(s)

Fabian Mueller (RnBeads)

df2granges *df2granges*

Description

Converts a data.frame that defines genomic regions to object of type GRanges.

Usage

```
df2granges(
  df,
  ids = rownames(df),
  chrom.col = 1L,
  start.col = 2L,
  end.col = 3L,
  strand.col = NULL,
  coord.format = "B1RI",
  assembly = NULL,
  doSort = FALSE,
  adjNumChromNames = FALSE
)
```

Arguments

df	Table defining genomic regions.
ids	Region names (identifiers) as a character vector, or NULL if no names are present.
chrom.col	Column name or index that lists the chromosome names.
start.col	Column name or index that lists the start positions of the regions.
end.col	Column name or index that lists the end positions of the regions.
strand.col	Column name or index that lists the strands on which the regions are located. Set this to NULL if this region set is not strand-specific.
coord.format	Coordinate format "B1RI" for 1-based right-inclusive (default), "B0RE" for 0-based right-exclusive.
assembly	Genome assembly of interest. See rnb.get.assemblies for the list of supported genomes.
doSort	Should the resulting table be sorted
adjNumChromNames	Should numeric chromosome names be adjusted for by adding the prefix "chr". Additionally chrMT becomes chrM. useful for converting GRC identifiers to NCBI identifiers

Value

GRanges object encapsulating of regions included in df. As GRanges, the coordinates will be 1-based right-inclusive. Columns other that the ones listed as parameters in this function are included as elementMetadata.

Examples

```
df <- data.frame(chrom=c(rep("chr5", 7), rep("chr21", 3)), start=1:10, end=seq(20, by=10, length.out=10), strand=r
df
df2granges(df, assembly="GRCh38_chr")
```

---

diagDivCellHeatmap *diagDivCellHeatmap*

Description

Plot a heatmap in which each cell is subdivided into a lower-left and upper-right triangle representing the values of two matrices

Usage

```
diagDivCellHeatmap(
  ml,
  mr,
  col.l = NULL,
  col.r = NULL,
  name.l = "Lower left",
  name.r = "Upper right",
  ...
)
```

Arguments

ml	the first value matrix (will be the left-lower diagonal in the cells of the resulting heatmap)
mr	the second value matrix (will be the right-upper diagonal in the cells of the resulting heatmap)
col.l	color scheme for the left-lower diagonal matrix. Should be generated by circlize::colorRamp2. Alternatively can be a character vector specifying color levels/breaks. If NULL a default color scheme will be used.
col.r	color scheme for the upper-right diagonal matrix. Should be generated by circlize::colorRamp2. Alternatively can be a character vector specifying color levels/breaks. If NULL a default color scheme will be used.
name.l	Name for the lower-left submatrix
name.r	Name for the upper-right submatrix
...	parameters passed on to ComplexHeatmap::Heatmap

Value

a ComplexHeatmap::Heatmap object containing the heatmap

Author(s)

Fabian Mueller

Examples

```
m1 <- matrix(rnorm(100, mean=0), ncol=10)
m2 <- matrix(rnorm(100, mean=2), ncol=10)
rownames(m1) <- rownames(m2) <- colnames(m1) <- colnames(m2) <- paste0("Idx", 1:10)
diagDivCellHeatmap(m1, m2, cluster_rows=FALSE, cluster_columns=FALSE)
cres <- as.hclust(muRtools::getClusteringDendrogram(m1, distMethod="euclidean", linkMethod="ward.D", corMethod="
diagDivCellHeatmap(m1, m2, cluster_rows=cres, cluster_columns=cres)
```

---

diagDivHeatmap *diagDivHeatmap*

Description

Plot a diagonally divided heatmap u

Usage

```
diagDivHeatmap(
  ml,
  mr,
  col.l = NULL,
  col.r = NULL,
  name.l = "Lower left",
  name.r = "Upper right",
  cluster = FALSE,
  cell.val.text = FALSE,
  cell.val.text.round = 2,
  ...
)
```

Arguments

ml	the first value matrix (the left-lower diagonal matrix will be in the result)
mr	the first value matrix (the right-upper diagonal matrix will be in the result)
col.l	color scheme for the left-lower diagonal matrix. Should be generated by circlize::colorRamp2. Alternatively can be a character vector specifying color levels/breaks. If NULL a default color scheme will be used.

col.r	color scheme for the upper-right diagonal matrix. Should be generated by circlize::colorRamp2. Alternatively can be a character vector specifying color levels/breaks. If NULL a default color scheme will be used.
name.l	Name for the lower-left submatrix
name.r	Name for the upper-right submatrix
cluster	logical or clustering object. will be passed to the cluster_rows and cluster_columns arguments of Heatmap
cell.val.text	logical indicating whether the cells value should be added as text
cell.val.text.round	if the cell value is numeric, the number of digits to which the cell text is rounded
...	parameters passed on to ComplexHeatmap::Heatmap

Value

a ComplexHeatmap::Heatmap object containing the heatmap

Author(s)

Fabian Mueller

Examples

```
m1 <- matrix(rnorm(100, mean=0), ncol=10)
m2 <- matrix(rnorm(100, mean=2), ncol=10)
rownames(m1) <- rownames(m2) <- colnames(m1) <- colnames(m2) <- paste0("Idx", 1:10)
diagDivHeatmap(m1, m2)
diagDivHeatmap(m1, m2, cell.val.text=TRUE, cell.val.text.round=3)
cres <- as.hclust(muRtools::getClusteringDendrogram(m1, distMethod="euclidean", linkMethod="ward.D", corMethod="
diagDivHeatmap(m1, m2, cluster=cres, cell.val.text=TRUE, cell.val.text.round=2)
```

---

dist.correlation *dist.correlation*

Description

Compute a distance matrix based on 1-corelation

Usage

```
dist.correlation(x, ...)
```

Arguments

x	a matrix on which the distances should be computed
...	parameters passed on to cor()

Value

a distance matrix

Author(s)

Fabian Mueller

`downloadLolaDbs` *downloadLolaDbs*

Description

Downloading prepared LOLA DBs from server

Usage

```
downloadLolaDbs(dest, dbs = c("LOLACore"))
```

Arguments

`dest`	destination directory
`dbs`	vector of names of LOLA DBs to be downloaded. Currently 'LOLACore' and 'LOLAExt' are supported

Details

Requires a stable internet connection. Could take a while depending on the size of the database and the internet connection

Value

a list containing vectors of directory names for each available genome assembly

Author(s)

Fabian Mueller

Examples

```
lolaDest <- tempfile()
dir.create(lolaDest)
lolaDirs <- downloadLolaDbs(lolaDest, dbs="LOLACore")
```

get.encode.cell.table *get.encode.cell.table*

Description

given the ENCODE controlled vocabulary file, retrieves a table characterizing the ENCODE cells

Usage

```
get.encode.cell.table(
  cvFile = "http://hgdownload.cse.ucsc.edu/goldenPath/encodeDCC/cv.ra"
)
```

Arguments

cvFile	the file location for the ENCODE controlled vocabulary file. Defaults to the one provided by ENCODE

Value

a table containing ENCODE cell annotations

Examples

```
ect <- get.encode.cell.table()
```

getAnnotGrl.gencode *getAnnotGrl.gencode*

Description

Create a GRangesList with element annotation by downloading the corresponding GTF file from Gencode

Usage

```
getAnnotGrl.gencode(name)
```

Arguments

name	gencode identifier. Currently supported are: "gencode.v27", "gencode.v19", "gencode.vM16", "gencode.vM1"

Value

GRangesList object with annotated elements for each element type (genes, transcripts, exons, ...)

Author(s)

Fabian Mueller

`getAssocTestRes.pca` *getAssocTestRes.pca*

Description

Test associations of annotations with principal components (PCA)

Usage

```
getAssocTestRes.pca(X, ph, nComp = 10, nPerm = 1000)
```

Arguments

`X`	A matrix on which the dimension reduction is to be performed. Alternatively, it can be a matrix of PC coordinates computed by `getDimRedCoords.pca`.
`ph`	annotation table for the datapoints. The columns of this table will be used to test associations with the PCs. Should be a `matrix` or `data.frame`.
`nComp`	number of PCs to be considered
`nPerm`	number of permutation tests to be conducted if an annotation in `ph` is numeric (i.e. a correlation permutation test is performed)

Value

A nested list of tested associations one element for each column in `ph` (1st level), each PC (2nd level). Each element is again a list with the name of the test being used (`test`), the test statistic (`statistic`) and p-value (`pvalue`)

Author(s)

Fabian Mueller

getCellTypesFromLolaDb *getCellTypesFromLolaDb*

Description

retrieve or guess cell types from a LOLA DB object

Usage

```
getCellTypesFromLolaDb(lolaDb)
```

Arguments

lolaDb	LOLA DB object as returned by LOLA::loadRegionDB or loadLolaDbs

Value

character vector with cell types

Author(s)

Fabian Mueller

Examples

```
# download LOLA DB
lolaDest <- tempfile()
dir.create(lolaDest)
lolaDirs <- downloadLolaDbs(lolaDest, dbs="LOLACore")
lolaDb <- loadLolaDbs(lolaDirs[["hg19"]])
getCellTypesFromLolaDb(lolaDb)
```

getClusteringDendrogram *getClusteringDendrogram*

Description

Get a clustering dendrogram using hierarchical clustering (wrapper)

Usage

```
getClusteringDendrogram(
  X,
  samplesOrdered = colnames(X),
  distMethod = "cor",
  linkMethod = "ward.D",
  corMethod = "pearson"
)
```

Arguments

X	A matrix for which the sample clustering dendrogram should be computed. Samples correspond to columns and features correspond to rows. Alternatively, X can be a distance matrix (dist object)
samplesOrdered	character vector specifying the preferred order of samples
distMethod	distance metric to be used for clusteing. must be either "cor" or a valid distance method for dist()
linkMethod	linkage method (see hclust for details)
corMethod	method for computing correlation coefficients. Only relevant if distMethod=="cor".

Value

clustering dendrogram (dendrogram object)

Author(s)

Fabian Mueller

---

getColorFun *getColorFun*

Description

Retrieve a color function mapping values to colors. Uses and modeled after circlize::colorRamp2.

Usage

```
getColorFun(x, colPal = NULL)
```

Arguments

x	vector or matrix including potential values
colPal	a color palette to be used. Should be a character vector specifying colors. can be named if specific colors should be used for specific values

Value

a function mapping a value to a corresponding color character string

Author(s)

Fabian Mueller

Examples

```
randomLetters <- sample(letters[1:6], 50, replace=TRUE)
cf_cat <- getColorFun(randomLetters)
cf_cat("f")
cf_cat("x") # NA
cf_cat2 <- getColorFun(randomLetters, colPal=c(a="#009FE3", b="#DE7E00", c="#8EC041", d="#FFCC00", e="#951B81", f
cf_num <- getColorFun(runif(50))
cf_num2 <- getColorFun(runif(50), colpal.cont(9, "cb.YlOrRd")))
cf_num2(0.5)
```

`getConfig` *getConfig*

Description

Get analysis configuration object from a JSON file. Automatically runs parsers for config elements specified in the '.MU_ANA_CONFIG' section of the JSON file

Usage

```
getConfig(cfgFn, anaName, addDirs = TRUE)
```

Arguments

`cfgFn`	JSON file name
`anaName`	name of the analysis to be conducted
`addDirs`	should the analysis directories be created and added to the object

Value

an S3 object containing analysis config elements

Author(s)

Fabian Mueller

getDimRedCoords.mds *getDimRedCoords.mds*

Description

Get dimension reduction coordinates (Multidimensional Scaling)

Usage

```
getDimRedCoords.mds(X, distMethod = "euclidean")
```

Arguments

X	A matrix on which the dimension reduction is to be performed
distMethod	distance metric to be employed

Value

a matrix containing two columns for the reduced dimensions and the same number of rows as X

Author(s)

Fabian Mueller

getDimRedCoords.pca *getDimRedCoords.pca*

Description

Get dimension reduction coordinates (PCA)

Usage

```
getDimRedCoords.pca(X, components = c(1, 2), method = "prcomp", ...)
```

Arguments

X	A matrix on which the dimension reduction is to be performed
components	principal component to be returned
method	Method/package to be used for computing principal components. Currently prcomp and irlba are supported.
...	other arguments passed on to the PCA method

Value

a matrix containing two columns for the reduced dimensions and the same number of rows as X

Author(s)

Fabian Mueller

`getDimRedCoords.tsne` *getDimRedCoords.tsne*

Description

Get dimension reduction coordinates (t-SNE)

Usage

```
getDimRedCoords.tsne(X, distMethod = "euclidean", dims = c(1, 2))
```

Arguments

`X`	A matrix on which the dimension reduction is to be performed
`distMethod`	distance metric to be employed
`dims`	dimensions to return from the reduction

Value

a matrix containing two columns for the reduced dimensions and the same number of rows as `X`

Author(s)

Fabian Mueller

`getDimRedCoords.umap` *getDimRedCoords.umap*

Description

Get dimension reduction coordinates using the UMAP method

Usage

```
getDimRedCoords.umap(X, distMethod = "euclidean", dims = c(1, 2), ...)
```

Arguments

`X`	A matrix on which the dimension reduction is to be performed
`distMethod`	distance metric to be employed
`dims`	dimensions to return from the reduction
`...`	parameters passed on to `uwot::umap()`

Value

a matrix containing two columns for the reduced dimensions and the same number of rows as `X`

Author(s)

Fabian Mueller

`getDimRedPlot` *getDimRedPlot*

Description

Generate a plot from dimension reduction coordinates

Usage

```
getDimRedPlot(
  coords,
  annot = NULL,
  colorCol = NULL,
  shapeCol = NULL,
  colScheme = "[auto]",
  ptSize = 3,
  addLabels = FALSE,
  addDensity = FALSE,
  addVoronoi = FALSE,
  annot.text = NULL,
  orderCol = NULL,
  facetCols = NULL
)
```

Arguments

`coords`	dimension reduction coordinates
`annot`	annotation matrix with the same number of rows as `coord`
`colorCol`	name or index in the annotation matrix (`annot`) that should be used for coloring the points if `colorCol` not supplied but `annot` is supplied, it defaults to the first annotation column
`shapeCol`	name or index in the annotation matrix (`annot`) that should be used for point shapes if `shapeCol` not supplied but `annot` is supplied and has more than one column, it defaults to the second annotation column
`colScheme`	color sheme to be used in coloring the points. can be a character vector with the supplied colors. Alternatively, if it is a one-element character vector `"[auto]"` the color scheme will be selected automatically using `muRtools::ggAutoColorScale`. If `NULL`, ggplots default color scheme will be used.

ptSize	size of the points in the scatterplot
addLabels	should observation labels be added to each point
addDensity	should Gaussian Kernel density estimation be performed and the contour lines plotted for each color group
addVoronoi	should a Voronoi tessalation grid (based on colorCol) be added to the plot
annot.text	optional text to be added in the lower right corner of the plot
orderCol	name or index in the annotation matrix (annot) that should be used for ordering the points. If not NULL Points will be ordered increasingly by their value, i.e. higher-valued points are plottet over lower-valued points
facetCols	name (string) of columns to be used for faceting the resulting plot. Each facet will contain all the points not in the facet as grey points.

Value

a ggplot2 object containing the dimension reduction plot

Author(s)

Fabian Mueller

Examples

```
df <- data.frame(
 x = c(rnorm(20, mean=0, sd=0.2), rnorm(10, mean=1, sd=0.4), rnorm(15, mean=1, sd=0.2)),
 y = c(rnorm(20, mean=0, sd=0.2), rnorm(10, mean=1, sd=0.4), rnorm(15, mean=0.5, sd=0.3)),
 group = rep(c("group1", "group2", "group3"), times=c(20,10,15)),
 stringsAsFactors=FALSE
)
getDimRedPlot(df[,c("x", "y")], annot=df[,c("group"), drop=FALSE], colorCol="group")
getDimRedPlot(df[,c("x", "y")], annot=df[,c("group"), drop=FALSE], colorCol="group", addDensity=TRUE)
```

---

getGeneAnnotMap *getGeneAnnotMap*

Description

Get a mapping (e.g. of identifiers) and automatically select the correct AnnotationDbi database for a given assembly

Usage

```
getGeneAnnotMap(assembly, from = "ENSEMBL", to = "SYMBOL", multiMap = "paste")
```

Arguments

assembly	character string specifying the assembly
from	the column name that will be used as the key for the resulting map
to	the column name that will be used as the result for the resulting map
multiMap	character string specifying what to do if multiple mappings are found for a key by default (multiMap="paste") the results will be pasted into a single character string (separated by ';'). Other options include 'first' for just returning the first value or 'list' for returning a list of all values

Value

a named vector (or list depending on how the multiMap argument is chosen) providing a mapping

Author(s)

Fabian Mueller

---

getGenomeGr *getGenomeGr*

Description

retrieve the full genome as GRanges object

Usage

```
getGenomeGr(assembly, ...)
```

Arguments

assembly	assembly
...	other arguments passed on to setGenomeProps

Value

GRanges object

getGenomeObject *getGenomeObject*

Description

retrieve the appropriate BSgenome for an assembly string

Usage

```
getGenomeObject(assembly, adjChrNames = TRUE)
```

Arguments

assembly	string specifying the assembly
adjChrNames	should the prefix "chr" be added to main chromosomes if not already present and chrMT be renamed to chrM?

Value

BSgenome object

getHashString *getHashString*

Description

Get a hash string, i.e. a string unlikely to occur again

Usage

```
getHashString(pattern = "", useDate = TRUE)
```

Arguments

pattern	a prefix that will be used in the returned hash string
useDate	Should the current time and date be used in the hash string to make it even more unique

Value

a character string unlikely to occur again

Author(s)

Fabian Mueller

Examples

```
getHashString()
```

`getNamesFromLolaDb` *getNamesFromLolaDb*

Description

get human readable names from a LOLA DB object

Usage

```
getNamesFromLolaDb(lolaDb, addCollectionNames = FALSE, addDbId = TRUE)
```

Arguments

`lolaDb`	LOLA DB object as returned by `LOLA::loadRegionDB` or `loadLolaDbs`
`addCollectionNames`	attach the name of the collection to the name
`addDbId`	attach the index of the item in the LOLA DB object to the name

Value

character vector with human readable names

Author(s)

Fabian Mueller

Examples

```
# download LOLA DB
lolaDest <- tempfile()
dir.create(lolaDest)
lolaDirs <- downloadLolaDbs(lolaDest, dbs="LOLACore")
lolaDb <- loadLolaDbs(lolaDirs[["hg19"]])
getNamesFromLolaDb(lolaDb)
```

getPointDensity *getPointDensity*

Description

Get point density of points in 2 dimensions. Code from http://slowkow.com/notes/ggplot2-color-by-density/

Usage

```
getPointDensity(x, y, n = 100)
```

Arguments

x	A numeric vector.
y	A numeric vector.
n	Create a square n by n grid to compute density.

Value

The density within each square

getRegionSet *getRegionSet*

Description

retrieves custom region sets by id

Usage

```
getRegionSet(id, assembly)
```

Arguments

id	region set id
assembly	genome assembly

Value

a GRanges object containing the region set

Examples

```
regs <- getRegionSet("deep_chip_ctrl_regions","hg19")
```

getRelatedAnaDirFromConfig
getRelatedAnaDirFromConfig

Description

Get analysis configuration object from a JSON file. Automatically runs parsers for config elements specified in the '.MU_ANA_CONFIG' section of the JSON file

Usage

```
getRelatedAnaDirFromConfig(cfg, anaName, anaVersion = cfg[[".anaVersion"]])
```

Arguments

cfg	configuration object as returned by getConfig
anaName	name of the analysis
anaVersion	version of the analysis. If NULL, it will look for the most recent one

Value

path of the related analysis directory

Author(s)

Fabian Mueller

getSeqlengths4assembly
getSeqlengths4assembly

Description

retrieve chromosomes/contigs and their sequence lengths for known assemblies

Usage

```
getSeqlengths4assembly(assembly, onlyMainChrs = FALSE, adjChrNames = TRUE)
```

Arguments

assembly	assembly
onlyMainChrs	should only main chromosomes, i.e. chr[1-N] + chr[XYM] be returned (e.g. not ChrUn*, *_random, ...)
adjChrNames	should the prefix "chr" be added to main chromosomes if not already present and chrMT be renamed to chrM?

Value

named vector of chromosomes/contigs and sequence lengths

`getTargetFromLolaDb` *getTargetFromLolaDb*

Description

retrieve or guess the target from a LOLA DB object. Here, target typically refers to antibodies for ChIP-seq experiments, but could also refer to other annotations (e.g. motifs in TF motif databases, annotation according to UCSC features etc.)

Usage

```
getTargetFromLolaDb(lolaDb)
```

Arguments

`lolaDb`	LOLA DB object as returned by `LOLA::loadRegionDB` or `loadLolaDbs`

Value

character vector with targets

Author(s)

Fabian Mueller

Examples

```
# download LOLA DB
lolaDest <- tempfile()
dir.create(lolaDest)
lolaDirs <- downloadLolaDbs(lolaDest, dbs="LOLACore")
lolaDb <- loadLolaDbs(lolaDirs[["hg19"]])
getTargetFromLolaDb(lolaDb)
```

getTilingRegions *getTilingRegions*

Description

Get a GRanges object of tiling regions for a specified genome assembly

Usage

```
getTilingRegions(assembly, width = 1000L, ...)
```

Arguments

assembly	assembly
width	tiling window size
...	arguments passed on to getSeqlengths4assembly

Value

GRanges object containing tiling windows

getTxDb.gencode *getTxDb.gencode*

Description

Create a TxDb object by downloading the corresponding GTF file from Gencode

Usage

```
getTxDb.gencode(name)
```

Arguments

name	gencode identifier. Currently supported are: "gencode.v27", "gencode.v19", "gencode.vM16", "gencode.vM1"

Value

TxDb object

Author(s)

Fabian Mueller

ggAutoColorScale *ggAutoColorScale*

Description

Automatical color scales for values for ggplots

Usage

```
ggAutoColorScale(x, method = "color", symmetric = TRUE)
```

Arguments

x	vector of values
method	method for scaling: "color" or "fill"
symmetric	treat numeric values as symmetric. If there are values smaller and larger than 0, a diverging color scheme will be applied

Value

the theme structure

Examples

```
dframe.num.pos <- data.frame(x=runif(100),y=runif(100))
ggplot(dframe.num.pos, aes(x=x,y=y, color=x)) + geom_point() + ggAutoColorScale(dframe.num.pos[,"x"])
dframe.num.sym <- data.frame(x=rnorm(100),y=rnorm(100))
ggplot(dframe.num.sym, aes(x=x,y=y, color=x)) + geom_point() + ggAutoColorScale(dframe.num.sym[,"x"])
dframe.num.sym.lab <- data.frame(x=rnorm(100),y=rnorm(100), lab=sample(c("A", "B", "C", "D"), 100, replace=TRUE))
ggplot(dframe.num.sym.lab, aes(x=x,y=y, color=lab)) + geom_point() + ggAutoColorScale(dframe.num.sym.lab[,"lab"]
```

ggMessagePlot *ggMessagePlot*

Description

Creates a plot, using **ggplot2**, with a single text message.

Usage

```
ggMessagePlot(txt)
```

Arguments

txt	Text to be plotted.

Value

The newly initialized ggplot instance.

Examples

```
ggMessagePlot("Missing data")
```

---

ggplot2.distr *ggplot2.distr*

Description

Distribution plot combining violin and boxplot using ggplot2

Usage

```
ggplot2.distr(x, fillColor = "#676D8D")
```

Arguments

x	vector of values whose distribution is to be plottet
fillColor	color to be used to fill the violin

Value

the ggplot2 object (can be extended for plotting)

Examples

```
x <- rnorm(1000)
ggplot2.distr(x)
```

---

ggplot2.heatmap *ggplot2.heatmap*

Description

converts a matrix or dataframe into a ggplot2 object for subsequent plotting.

Usage

```
ggplot2.heatmap(mm, add.text = FALSE)
```

Arguments

mm	matrix or dataframe to be plottet as heatmap
add.text	logical. should the cells be labelled with the values

Value

the ggplot2 object (can be extended for plotting)

Examples

```
ggplot2.heatmap(airquality[1:15,],add.text=TRUE) + scale_fill_gradient(low = "red",high = "steelblue"))
```

---

ggsave4doc *ggsave4doc*

Description

Wrapper around ggsave that has default values for parameters fitting for embedding plots into my documents

Usage

```
ggsave4doc(
  fn,
  plot = last_plot(),
  width = 192,
  height = 192,
  units = "mm",
  family = "Helvetica",
  dimPreset = NULL,
  useDingbats = FALSE,
  ...
)
```

Arguments

fn	file name
plot	see ?ggsave
width	see ?ggsave
height	see ?ggsave
units	see ?ggsave
family	see ?ggsave
dimPreset	presets for figure dimensions. Possible values are NULL (don't use a preset; default), "slide_nuc_wide_full" (full slide using Fabian's wide nucleosome template), "slide_nuc_wide_half" (half a slide using Fabian's wide nucleosome template) Overwrites width, height and units.

useDingbats	see ?ggsave
...	see ?ggsave

Value

result of ggsave command

---

ggtemp *ggtemp*

Description

Wrapper for quickly saving plot to temporary file

Usage

```
ggtemp(
  plot = last_plot(),
  fn = paste0("~/tmp_work/", getHashString("ggplot"), ".pdf"),
  ...
)
```

Arguments

plot	see ?ggsave
fn	file name
...	see ?ggsave

Value

result of ggsave command

---

goEnrichment *goEnrichment*

Description

Perform Gene Ontology (GO) enrichment using the topGO package

Usage

```
goEnrichment(
  qids,
  uids,
  ontology = "BP",
  idType = "ensembl",
  assembly = "org.Hs.eg.db",
  algorithm = "weight01"
)
```

Arguments

qids	character vector of query gene IDs
uids	character vector of universe gene IDs
ontology	character specifying the ontology to use (default: "BP")
idType	character specifying which universe the gene IDs come from (default: "ensembl"). Possible values are: "entrez", "genbank", "alias", "ensembl", "symbol", "genename", "unigene"
assembly	character specifying the genome to use. Can either be the name of the package to be used for mapping the identifiers (e.g. "org.Hs.eg.db"; default) or an identifier for a genome assembly ((e.g. "hg38")
algorithm	algorithm employed by topGO. See topgGO::runTest for details.

Value

a list (S3 class) object containing: - $tgData: the used topGOdata object - $resultObj: the resulting topGOresult object - $table: a summary table of statistics for each GO term

Author(s)

Fabian Mueller

---

granges2bed *granges2bed*

Description

Save a GRanges object to a bed file

Usage

```
granges2bed(
  gr,
  fn,
  score = NULL,
  addAnnotCols = FALSE,
  colNames = FALSE,
  doSort = TRUE,
  bedgraph = FALSE,
  bigBed = FALSE,
  tabix = FALSE,
  strandCharNA = ".",
  coordOnly = FALSE
)
```

Arguments

gr	GRanges object
fn	filename to save bed file to
addAnnotCols	add the columns stored in elementMetadata of GRanges
colNames	add column names
doSort	sort the regions before writing the output
bedgraph	export to bedgraph instead of bed
bigBed	also save as bigbed file. Requires that the GRanges object has chromosome sizes stored.
tabix	compress and index by tabix
strandCharNA	character to be used if strand is NA, '*' or '.'
coordOnly	output only the coordinates and strand information (only taken into account if addAnnotCols==FALSE). If all strand information is NA, it will be dropped as well.
sc	score vector or column in elementMetadata of GRanges

Value

(invisibly) the written results as a data.frame

---

granges2bed.igv *granges2bed.igv*

Description

Save a GRanges object to a bed file which can be displayed by IGV

Usage

```
granges2bed.igv(
  gr,
  fn,
  trackName = NULL,
  scoreCol = NULL,
  na.rm = FALSE,
  nameCol = NULL,
  col.cat = colpal.bde,
  col.cont = c("#EDF8B1", "#41B6C4", "#081D58"),
  col.na = "#bdbdbd",
  col.range = NULL,
  doSort = TRUE
)
```

Arguments

gr	GRanges object
fn	filename to save bed file to
trackName	track name to be displayed
scoreCol	the score column (in the GRanges elementMetadata) that is optionally used for coloring
na.rm	flag indicating whether items with NA score should be removed
nameCol	the name column (in the GRanges elementMetadata) that is used for labelling the items
col.cat	color panel for coloring categorical scores
col.cont	color panel for coloring numerical scores
col.na	color used for NA scores
col.range	vector of length 2 indicating the range of scores for the color scales to be applied (continuous scores only)
doSort	sort the regions before writing the output

Value

invisibly, the resulting data frame containing the bed file columns

---

`granges2igv` *granges2igv*

Description

Save a GRanges object to a IGV file

Usage

```
granges2igv(
  gr,
  fn,
  addStrand = FALSE,
  addAnnotCols = TRUE,
  doSort = TRUE,
  toTDF = FALSE
)
```

Arguments

gr	GRanges object
fn	filename to save IGV file to
addAnnotCols	add the columns stored in elementMetadata of GRanges
doSort	sort the regions before writing the output
toTDF	convert to TDF file. Requires that "igvtools" is executable from the current path
sc	score vector or column in elementMetadata of GRanges

Value

result of writing the table (see `write.table`)

`grGeneAnnot` *grGeneAnnot*

Description

get gene annotation for a `GRanges` object using a `RegionSetDB` region database object by linking to the nearest gene

Usage

```
grGeneAnnot(
  gr,
  rsdb,
  geneSetName = "genes_protein_coding",
  geneSetCollection = "Gencode",
  maxDist = 1e+05
)
```

Arguments

gr	GRanges object to liftOver
rsdb	RegionSetDB object containing a region set database from which gene annotation can be retrieved
geneSetName	Name of the region set containng gene annotation in the RegionSetDB
geneSetCollection	Name of the region set collection containng gene annotation in the RegionSetDB
maxDist	maximum distance for matching to nearest gene

Value

data.frame containing information on the nearest gene for each element in gr

---

grLiftOver *grLiftOver*

Description

Converts coordinates of a GRanges object to target genome assembly. Wraps around rtracklayer::liftOver and automatically downloads and selects the correct chain file

Usage

```
grLiftOver(gr, targetAssembly, onlyUnique = TRUE)
```

Arguments

gr	GRanges object to liftOver
targetAssembly	character string specifying the target assembly

Value

GRanges object with coordinates that could uniquely be

grSignedDistance *grSignedDistance*

Description

Compute pairwise distances between the elements of two GRanges objects, taking orientation and position into account. (wrapper for GRanges::distance)

Usage

```
grSignedDistance(gr1, gr2)
```

Arguments

gr1	GRanges object 1
gr2	GRanges object 2

Value

vector of pairwise distances Elements in which the region in gr2 is upstream of the region in gr1 will be assigned negative distances. "Upstream" is defined based on the orientation of the regions in gr1.

grTile *grTile*

Description

Tile each element in a GRanges object into equally-sized windows. If the length of an element is not divisible by the window-size, each element will be adjusted to match a multiple of the desired window-size

Usage

```
grTile(gr, tile.width = 200, keepMetadata = TRUE)
```

Arguments

gr	GRanges object to liftOver
tile.width	length of the tiling window
keepMetadata	Should the metadata columns for each element be preserved in the resulting object

Value

GRanges containing the tiling regions. Additional metadata columns named .orgIdx, .winIdx denote the indices of the original element and the window respectively

indicesInList *indicesInList*

Description

Find the occurrences of items in a list of vectors

Usage

```
indicesInList(x, l)
```

Arguments

x	vector of items to be found in the list
l	list of vectors in which x should be found

Value

a list containing an element for each item in x that contains the indices of its occurrence in l

Examples

```
l <- list(1:3, 4:5, 5:9)
x <- c(2,3,5,666,8,5)
indicesInList(x, l)
```

kde.plot.simple *kde.plot*

Description

plot a single vector as kernel density estimation (kde.plot.simple), plot columns of a matrix as kernel density estimation (kde.plot.matrix)

Usage

```
kde.plot.simple(x, initial = FALSE, col = c("#00640044"), ...)

kde.plot.matrix(X, col = makeTrans(rainbow(ncol(X))), legend = TRUE, ...)
```

Arguments

x	values
initial	is the plot initial, i.e. should the plot() function be called
col	color(s)
...	more graphical parameters
X	matrix
legend	add a color legend with colnames of the matrix

Value

Nothing particularly interesting

Examples

```
kde.plot.simple(c(1:10,rep(5,3)),initial=TRUE,col=c("#00640044"),main="Some plot")
kde.plot.simple(rep(8,2),initial=FALSE,col=c("#64640044"))

dd <- USJudgeRatings
dd[,5] <- 1/dd[,5]
tt <- dd[,1:5]
kde.plot.matrix(tt)
```

`loadLolaDbs` *loadLolaDbs*

Description

Load LOLA databases from disk and merge them

Usage

```
loadLolaDbs(lolaDbPaths, collections = NULL)
```

Arguments

`lolaDbPaths`	vector of names of LOLA DB paths to be loaded
`collections`	Restrict the database loading to this list of collections. passed to `LOLA::loadRegionDB`

Value

LOLA DB list as returned by `LOLA::loadRegionDB`

Author(s)

Fabian Mueller

Examples

```
# download LOLA DB
lolaDest <- tempfile()
dir.create(lolaDest)
lolaDirs <- downloadLolaDbs(lolaDest, dbs="LOLACore")
lolaDb <- loadLolaDbs(lolaDirs[["hg19"]])
```

loadRnBeadsAnalysis *loadRnBeadsAnalysis*

Description

Loads RnBeads analysis results (RnBSet object, differential methylation) and optionally runs the corresponding preanalysis script and sets options

Usage

```
loadRnBeadsAnalysis(
  input,
  type = "cluster_run",
  preprocessed = TRUE,
  setOptions = TRUE,
  preAnalysis = TRUE,
  diffMeth = TRUE
)
```

Arguments

input	input directory
type	analysis type. Currently only "cluster_run" is supported
preprocessed	determines the type of RnBSet object to be loaded. If TRUE (default) the preprossed/filtered object will be loaded. Otherwise the raw, imported object
setOptions	should the analysis options be set from the options settings of the RnBeads run?
preAnalysis	should the corresponding preanalysis script be called
diffMeth	should the differential analysis result be loaded in addition?

Value

A list containing the RnBSet object (list item rnbs) and optionally the differential methylaiton result (list item diffmeth)

Author(s)

Fabian Mueller

lolaBarPlot *lolaBarPlot*

Description

plot a barplot of LOLA enrichment results

Usage

```
lolaBarPlot(
  lolaDb,
  lolaRes,
  scoreCol = "pValueLog",
  orderCol = scoreCol,
  signifCol = "qValue",
  includedCollections = c(),
  recalc = TRUE,
  pvalCut = 0.01,
  maxTerms = 50,
  colorpanel = sample(rainbow(maxTerms, v = 0.5)),
  groupByCollection = TRUE,
  orderDecreasing = NULL,
  appendTermDbId = TRUE
)
```

Arguments

lolaDb	LOLA DB object as returned by LOLA::loadRegionDB or loadLolaDbs
lolaRes	LOLA enrichment result as returned by the runLOLA function from the LOLA package
scoreCol	column name in lolaRes to be plotted
orderCol	column name in lolaRes which is used for sorting the results
signifCol	column name of the significance score in lolaRes. Should be one of c("pValueLog", "qValue")
includedCollections	vector of collection names to be included in the plot. If empty (default), all collections are used
recalc	recalculate adjusted p-value/q-value and ranks after the specified subsetting (by includedCollections)
pvalCut	p-value cutoff to be employed for filtering the results
maxTerms	maximum number of items to be included in the plot
colorpanel	colors to be used for coloring the bars according to "target" (see getTargetFromLolaDb). An empty vector indicates that black will be used for all bars.
groupByCollection	facet the plot by collection

`orderDecreasing` flag indicating whether the value in `orderCol` should be considered as decreasing (as opposed to increasing). `NULL` (default) for automatic determination.

`appendTermDbId` attach the index of the item in the LOLA DB object to the name of the set

Value

ggplot object containing the plot

Author(s)

Fabian Mueller

Examples

```
# example taken from RnBeads
library(RnBeads.hg19)
data(small.example.object)
logger.start(fname=NA)
# compute differential methylation
dm <- rnb.execute.computeDiffMeth(rnb.set.example,pheno.cols=c("Sample_Group","Treatment"))
# download LOLA DB
lolaDest <- tempfile()
dir.create(lolaDest)
lolaDirs <- downloadLolaDbs(lolaDest, dbs="LOLACore")
# perform enrichment analysis
res <- performLolaEnrichment.diffMeth(rnb.set.example,dm,lolaDirs[["hg19"]])
# select the 500 most hypermethylated tiling regions in ESCs compared to iPSCs
# in the example dataset
lolaRes <- res$region[["hESC vs. hiPSC (based on Sample_Group)"]][["tiling"]]
lolaRes <- lolaRes[lolaRes$userSet=="rankCut_500_hyper",]
# plot
lolaBarPlot(res$lolaDb, lolaRes, scoreCol="oddsRatio", orderCol="maxRnk", pvalCut=0.05)
```

---

lolaBoxPlotPerTarget *lolaBoxPlotPerTarget*

Description

plot a boxplot showing LOLA enrichment results per "target" group (see `getTargetFromLolaDb` for an explanation of "target").

Usage

```
lolaBoxPlotPerTarget(
  lolaDb,
  lolaRes,
  scoreCol = "pValueLog",
```

```
  orderCol = scoreCol,
  signifCol = "qValue",
  includedCollections = c(),
  recalc = TRUE,
  pvalCut = 0.01,
  maxTerms = 50,
  colorpanel = c(),
  groupByCollection = TRUE,
  orderDecreasing = NULL,
  scoreDecreasing = NULL
)
```

Arguments

lolaDb	LOLA DB object as returned by LOLA::loadRegionDB or loadLolaDbs
lolaRes	LOLA enrichment result as returned by the runLOLA function from the LOLA package
scoreCol	column name in lolaRes to be plotted
orderCol	column name in lolaRes which is used for sorting the results
signifCol	column name of the significance score in lolaRes. Should be one of c("pValueLog", "qValue")
includedCollections	vector of collection names to be included in the plot. If empty (default), all collections are used
recalc	recalculate adjusted p-value/q-value and ranks after the specified subsetting (by includedCollections)
pvalCut	p-value cutoff to be employed for filtering the results
maxTerms	maximum number of items to be included in the plot
colorpanel	colors to be used for coloring the bars according to "target" (see getTargetFromLolaDb). An empty vector indicates that black will be used for all bars.
groupByCollection	facet the plot by collection
orderDecreasing	flag indicating whether the value in orderCol should be considered as decreasing (as opposed to increasing). NULL (default) for automatic determination.
scoreDecreasing	flag indicating whether the value in scoreCol should be considered as decreasing (as opposed to increasing). NULL (default) for automatic determination.

Value

ggplot object containing the plot

Author(s)

Fabian Mueller

Examples

```
# example taken from RnBeads
library(RnBeads.hg19)
data(small.example.object)
logger.start(fname=NA)
# compute differential methylation
dm <- rnb.execute.computeDiffMeth(rnb.set.example,pheno.cols=c("Sample_Group","Treatment"))
# download LOLA DB
lolaDest <- tempfile()
dir.create(lolaDest)
lolaDirs <- downloadLolaDbs(lolaDest, dbs="LOLACore")
# perform enrichment analysis
res <- performLolaEnrichment.diffMeth(rnb.set.example,dm,lolaDirs[["hg19"]])
# select the 500 most hypermethylated tiling regions in ESCs compared to iPSCs
# in the example dataset
lolaRes <- res$region[["hESC vs. hiPSC (based on Sample_Group)"]][["tiling"]]
lolaRes <- lolaRes[lolaRes$userSet=="rankCut_500_hyper",]
# plot
lolaBoxPlotPerTarget(res$lolaDb, lolaRes, scoreCol="oddsRatio", orderCol="maxRnk", pvalCut=0.05)
```

---

lolaRegionSetHeatmap *lolaRegionSetHeatmap Plot a heatmap in which the rows are different user sets and the color corresponds to an enrichment score*

Description

lolaRegionSetHeatmap Plot a heatmap in which the rows are different user sets and the color corresponds to an enrichment score

Usage

```
lolaRegionSetHeatmap(
  lolaDb,
  lolaRes,
  scoreCol = "pValueLog",
  orderCol = scoreCol,
  signifCol = "qValue",
  markSignif = FALSE,
  includedCollections = c(),
  recalc = TRUE,
  pvalCut = 0.01,
  maxTerms = 50,
  userSetOrder = NULL,
  colorpanel = colpal.cont(9, "cb.OrRd"),
  colorpanelLimits = rep(as.numeric(NA), 2),
  groupByCollection = TRUE,
  orderDecreasing = NULL,
```

```
  appendTermDbId = TRUE
)
```

Arguments

lolaDb	LOLA DB object as returned by LOLA::loadRegionDB or loadLolaDbs
lolaRes	LOLA enrichment result as returned by the runLOLA function from the LOLA package
scoreCol	column name in lolaRes to be plotted
orderCol	column name in lolaRes which is used for sorting the results
signifCol	column name of the significance score in lolaRes. Should be one of c("pValueLog", "qValue")
markSignif	mark significant enrichments in the heatmap
includedCollections	vector of collection names to be included in the plot. If empty (default), all collections are used
recalc	recalculate adjusted p-value/q-value and ranks after the specified subsetting (by includedCollections)
pvalCut	p-value cutoff (negative log10) to be employed for filtering the results
maxTerms	maximum number of items to be included in the plot
userSetOrder	the order in which the user sets are displayed. NULL (default) means original factor levels if userSet is a factor or (alphanumeric) sorting otherwise; "nSignif" means in decreasing order of significant terms; "clusterSignif" means hierarchical clustering by occurrences of significant terms;
colorpanel	colorpanel for the heatmap gradient
groupByCollection	facet the plot by collection
orderDecreasing	flag indicating whether the value in orderCol should be considered as decreasing (as opposed to increasing). NULL (default) for automatic determination.
appendTermDbId	attach the index of the item in the LOLA DB object to the name of the set

Value

ggplot object containing the plot

Author(s)

Fabian Mueller

lolaVolcanoPlot *lolaVolcanoPlot*

Description

plot a volcano plot showing LOLA enrichment results: LOLA p-value against the log-odds score. Colored by rank

Usage

```
lolaVolcanoPlot(
  lolaDb,
  lolaRes,
  includedCollections = c(),
  signifCol = "qValue",
  recalc = TRUE,
  colorBy = "maxRnk",
  colorpanel = c()
)
```

Arguments

lolaDb	LOLA DB object as returned by LOLA::loadRegionDB or loadLolaDbs
lolaRes	LOLA enrichment result as returned by the runLOLA function from the LOLA package
includedCollections	vector of collection names to be included in the plot. If empty (default), all collections are used
signifCol	column name of the significance score in lolaRes. Should be one of c("pValueLog", "qValue").
recalc	recalculate adjusted p-value/q-value and ranks after the specified subsetting (by includedCollections)
colorBy	annotation/column in the the LOLA DB that should be used for point coloring
colorpanel	colors to be used for coloring the points

Value

ggplot object containing the plot

Author(s)

Fabian Mueller

Examples

```
# example taken from RnBeads
library(RnBeads.hg19)
data(small.example.object)
logger.start(fname=NA)
# compute differential methylation
dm <- rnb.execute.computeDiffMeth(rnb.set.example,pheno.cols=c("Sample_Group","Treatment"))
# download LOLA DB
lolaDest <- tempfile()
dir.create(lolaDest)
lolaDirs <- downloadLolaDbs(lolaDest, dbs="LOLACore")
# perform enrichment analysis
res <- performLolaEnrichment.diffMeth(rnb.set.example,dm,lolaDirs[["hg19"]])
# select the 500 most hypermethylated tiling regions in ESCs compared to iPSCs
# in the example dataset
lolaRes <- res$region[["hESC vs. hiPSC (based on Sample_Group)"]][["tiling"]]
lolaRes <- lolaRes[lolaRes$userSet=="rankCut_500_hyper",]
# plot
lolaVolcanoPlot(res$lolaDb, lolaRes, signifCol="qValue")
```

`makeTrans` *makeTrans*

Description

make a vector of colors transparent

Usage

```
makeTrans(ccc, isText = FALSE, transparancy.val = "44")
```

Arguments

`ccc`	color vector to make transparent
`isText`	are the colors textstrings (as opposed to hex value strings)?
`transparancy.val`	string value for transparancy

Value

a vector of colors with transparancy (hex strings)

Examples

```
makeTrans(rainbow(6))
makeTrans(c("red","dark blue","coral"),isText=TRUE)
```

matchStrand *matchStrand*

Description

match commonly used strand names to "+", "-", "*"

Usage

```
matchStrand(values)
```

Arguments

values	character vector or factor of strand names

Value

Factor of genomic strand (with levels "+", "-", "*")

muRtools *muRtools: Mueller's R tools*

Description

Fabian's custom plotting functions and utilities

normalize.str *normalize.str*

Description

normalize a string by removing special characters and replacing whitespaces and dots and afterwards remove all leading and trailing whitespaces and special characters By default, underscore is the replacement character. Avoids consecutive underscores.

Usage

```
normalize.str(x, resolve.camel = FALSE, return.camel = FALSE)
```

Arguments

resolve.camel	Is the string in camelCase and should this be resolved to snake_case?
return.camel	should camelCase be outputed rather than undescores?
a	string or string vector

Value

the normalized string(s)

Examples

```
normalize.str("_ (b)lA BLu[bb.blA\tblubb- bla)\n_")
```

---

normalizePercentile *normalizePercentile*

Description

Performs percentile normalization on the columns of a matrix, i.e. each element in a column will be the percentile it lies in in its column

Usage

```
normalizePercentile(X)
```

Arguments

X	A matrix which should be normalized

Value

a matrix containing the normalized values

Author(s)

Fabian Mueller

---

normalizeRank *normalizeRank*

Description

Performs rank normalization on the columns of a matrix

Usage

```
normalizeRank(X, out = "rank", ties.method = "average")
```

Arguments

X	A matrix which should be normalized
out	output type. Either "rank" or "percentile"
ties.method	method for breaking ties (see ?colRanks for details)

Value

a matrix containing the normalized values

Author(s)

Fabian Mueller

`pairsDensCor` *pairsDensCor*

Description

Pair plot with correlation values colored and resized on the upper right diagonal and density scatter plot on the lower left diagonal

Usage

```
pairsDensCor(
  tt,
  colscheme = c(gplots::colorpanel(100, "blue", "white"), gplots::colorpanel(100,
    "white", "red")),
  ...
)
```

Arguments

`tt`	table to be visualized
`colscheme`	colorscheme to be used for correlation
`...`	more plotting parameters

Value

Nothing particularly interesting

Examples

```
dd <- USJudgeRatings
dd[,5] <- 1/dd[,5]
tt <- dd[,1:5]
pairsDensCor(tt)
```

panel.cor.col *panel.cor.col*

Description

creates a correlation panel. The correlation values are colored and fontsize is proportional to amount of correlation

Usage

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

Arguments

x	x
y	y
digits	number of digits to be displayed
prefix	prefix for correlation text
cex.cor	cex for correlation text
...	more plotting parameters

Value

Nothing particularly interesting

panel.density *panel.density*

Description

creates a scatterplot density panel.

Usage

```
panel.density(x, y, ...)
```

Arguments

x	x
y	y
...	more plotting parameters

Value

Nothing particularly interesting

parse.cl.args *parse.cl.args*

Description

parser for command line arguments

Usage

```
parse.cl.args()
```

Value

a named list with command line arguments

Examples

```
cmd.args <- parse.cl.args()
```

parse.encode.cv.file *parse.encode.cv.file*

Description

parser for the ENCODE projects controlled vocabulary file

Usage

```
parse.encode.cv.file(
  cvFile = "http://hgdownload.cse.ucsc.edu/goldenPath/encodeDCC/cv.ra"
)
```

Arguments

cvFile	the file location for the ENCODE controlled vocabulary file. Defaults to the one provided by ENCODE

Value

a list containing blank line seperated blocks in each element. Each element is a named list containing the content to each keyword. a keyword is the first word in a line.

Examples

```
cv.blocks <- parse.cv.file()
```

pdftemp *pdftemp*

Description

Wrapper for quickly saving plot to temporary pdf file. terminate using `dev.off()`

Usage

```
pdftemp(fn = paste0("~/tmp_work/", getHashString("rplot"), ".pdf"), ...)
```

Arguments

`fn`	file name
`...`	see `?pdf`

Value

nothing of particular interest

plotAllDimRed *plotAllDimRed*

Description

Generate a plots with multiple methods and parameter settings from a feature matrix

Usage

```
plotAllDimRed(
  X,
  fn.prefix = NULL,
  fn.suffix = "",
  annot = NULL,
  distMethods = c(euc = "euclidean", man = "manhattan"),
  width = 10,
  height = 10,
  ...
)
```

Arguments

X	feature matrix containing one row for each observation and one column for each feature
fn.prefix	file prefix to be used for the resulting plots. If NULL, no plot is actually created, but a list of resulting plot objects is returned
fn.suffix	file suffix to be used for the resulting plots
annot	annotation matrix with the same number of rows as X
distMethods	distance methods for MDS and t-SNE
width	width of the resulting plot
height	height of the resulting plot
...	arguments to be passed on to getDimRedPlot

Details

Currently, PCA, MDS and t-SNE are employed by default with euclidean and manhattan distance metrics where applicable

Value

(invisibly) a list of lists containing the created plots as ggplot objects and additional info for each plot

Author(s)

Fabian Mueller

---

plotColpal *plotColpal*

Description

Get a continuous color palette

Usage

```
plotColpal(cp, type = "pie")
```

Arguments

cp	color palette, i.e. vector of colors
type	pie chart or stripes

Value

nothing of particular interest

Author(s)

Fabian Mueller

`plotCorPhm` *plotCorPhm*

Description

Plot a correlation matrix as heatmap. Wraps around `pheatmap`. Note that no clustering will be performed if not supplied with an appropriate clustering dendrogram

Usage

```
plotCorPhm(
  cc,
  clustDend = NULL,
  sampleAnnot = NA,
  color = colorRampPalette(rev(colpal.cont(n = 11, name = "cb.RdBu")))(100),
  breaks = seq(-1 - 1e-06, 1 + 1e-06, length.out = length(color) + 1),
  border_color = NA,
  ...
)
```

Arguments

`cc`	a correlation matrix (as returned by `cor()`)
`clustDend`	a clustering dendrogram to be used. Set to `NULL` to disable clustering dendrogram
`sampleAnnot`	a data.frame containing sample information to color by (corrsponds to `annotation_row` and `annotation_col` parameters of `pheatmap()`)
`color`	see `?pheatmap` for details
`breaks`	see `?pheatmap` for details. In this wrapper, the default value corresponds to splitting color across the full range of correlation coefficients [-1,1]
`border_color`	see `?pheatmap` for details
`...`	parameters passed on to `pheatmap()`

Value

invisibly the result of a call to `pheatmap()`

Author(s)

Fabian Mueller

plotDimRed *getDimRedPlot*

Description

Generate a plot from a feature matrix

Usage

```
plotDimRed(
  X,
  dimRedFun = getDimRedCoords.pca,
  annot = NULL,
  colorCol = NULL,
  shapeCol = NULL,
  colScheme = NULL,
  ptSize = 3,
  addLabels = FALSE,
  addDensity = FALSE,
  annot.text = NULL,
  ...
)
```

Arguments

X	feature matrix containing one row for each observation and one column for each feature
dimRedFun	function to do dimension reduction. E.g. getDimRedCoords.pca, getDimRedCoords.mds, getDimRedCoords.tsne,
annot	annotation matrix with the same number of rows as X
colorCol	name or index in the annotation matrix (annot) that should be used for coloring the points if colorCol not supplied but annot is supplied, it defaults to the first annotation column
shapeCol	name or index in the annotation matrix (annot) that should be used for point shapes if shapeCol not supplied but annot is supplied and has more than one column, it defaults to the second annotation column
colScheme	color sheme to be used in coloring the points
ptSize	size of the points in the scatterplot
addLabels	should observation labels be added to each point
addDensity	should Gaussian Kernel density estimation be performed and the contour lines plotted for each color group
annot.text	optional text to be added in the lower right corner of the plot
...	arguments to be passed on to dimRedFun

Value

a ggplot2 object containing the dimension reduction plot

Author(s)

Fabian Mueller

---

plotFisherTest *plotFisherTest*

Description

Conduct a Fisher's exact test and plot the results as a heatmap

Usage

```
plotFisherTest(x, y = NULL, name.x = NULL, name.y = NULL, ...)
```

Arguments

x	factor object or one that can be coerced to one. Alternative a 2x2 contingency matrix
y	factor object or one that can be coerced to one
name.x	optional character string specifying the name for the first grouping
name.y	optional character string specifying the name for the second grouping
...	arguments passed on to fisher.test

Value

an S3 object containing the test result object as returned by fisher.test and a ggplot object

Author(s)

Fabian Mueller

pngtemp *pngtemp*

Description

Wrapper for quickly saving plot to temporary png file. terminate using `dev.off()`

Usage

```
pngtemp(
  fn = paste0("~/tmp_work/", getHashString("rplot"), ".png"),
  width = 1024,
  height = 1024,
  ...
)
```

Arguments

`fn`	file name
`...`	see ?png

Value

nothing of particular interest

randomGroupedHeatmap *randomGroupedHeatmap*

Description

generate a random grouped heatmap using `ComplexHeatmap`.

Usage

```
randomGroupedHeatmap(
  n.row = 20,
  n.col = 6,
  ngrps.row = 2,
  ngrps.col = 3,
  cols = NULL,
  ...
)
```

Arguments

n.row	number of rows
n.col	number of columns
ngrps.row	number of groups to group rows into
ngrps.col	number of groups to group columns into
cols	color scheme. Should be a color character vector. If NULL a default color scheme will be used.
...	parameters passed on to ComplexHeatmap::Heatmap

Value

a ComplexHeatmap::Heatmap object containing the heatmap

Author(s)

Fabian Mueller

Examples

```
randomGroupedHeatmap(n.row=10, n.col=3, ngrps.row=2, ngrps.col=3, cols=colpal.cont(n=9, name="viridis"))
randomGroupedHeatmap(n.row=100, n.col=18, ngrps.row=3, ngrps.col=3, cols=colpal.cont(n=9, name="cb.BrBG"))
pdftemp()
draw(randomGroupedHeatmap(n.row=100, n.col=6, ngrps.row=3, ngrps.col=3, cols=colpal.PhFr.a))
dev.off()
```

---

readTab *readTab*

Description

Wrapper around read.table to read tab-separated tables by default

Usage

```
readTab(
  fn,
  sep = "\t",
  header = TRUE,
  stringsAsFactors = FALSE,
  quote = "",
  comment.char = "",
  na.strings = "",
  ...
)
```

Arguments

fn	filename to read
sep	see ?read.table
header	see ?read.table
stringsAsFactors	see ?read.table
quote	see ?read.table
comment.char	see ?read.table
...	passed to read.table

Value

the result of read.table

Author(s)

Fabian Mueller

---

reloadPackage *reloadPackage*

Description

reloads a package without quitting the R session. Useful for developing packages interactively

Usage

```
reloadPackage(package.name)
```

Arguments

package.name	name (string) of the package to be reloaded

Value

result of library(package.name)

Examples

```
library(GenomicRanges)
reloadPackage("GenomicRanges")
```

`rowTtest` *rowTtest*

Description

performs a two-sided Welch's t-test (unequal variances, equal or unequal sample sizes) on each row of a matrix X with the indices inds.1 vs indices idx2 as group assignments.

Usage

```
rowTtest(X, idx1, idx2 = -idx1, na.rm = FALSE, alternative = "two.sided")
```

Arguments

`X`	Matrix on which the test is performed for every row
`idx1`	column indices of group 1 members
`idx2`	column indices of group 2 members
`na.rm`	Should NAs be removed (logical)
`alternative`	Testing alternative. Must be one of "two.sided" (default),"less","greater" or "all". in case of "all" a data frome with corresping alternative variables is returned. Otherwise the result is a vector.

Value

vector (or data.frame if alternative=="all") of p-values resulting from the Welch's t-test

Note

Requires `matrixStats` package

Author(s)

Fabian Mueller

`runLOLA_list` *runLOLA_list More robust version of lola for lists of user sets (to avoid "Negative c entry in table" errors)*

Description

runLOLA_list More robust version of lola for lists of user sets (to avoid "Negative c entry in table" errors)

Usage

```
runLOLA_list(userSets, userUniverse, lolaDb, ...)
```

Arguments

userSets	NAMED list or GRangesList of user sets
userUniverse	GRanges object to be used as universe (as required by LOLA::runLOLA)
lolaDb	LOLA DB object as returned by LOLA::loadRegionDB or loadLolaDbs
...	other arguments to iterate over

Value

LOLA result as returned by LOLA::runLOLA

Author(s)

Fabian Mueller

---

scatter.twogroups *scatter.twogroups*

Description

density scatterplot with highlighting the points of a certain group in a different color

Usage

```
scatter.twogroups(
  x,
  y,
  is.g,
  cols.all = blues9[-(1:3)],
  cols.g = c("coral", "darkred"),
  ...
)
```

Arguments

x	x coordinates
y	y coordinates
is.g	logical vector indicating whether a point is in the highlighted group
cols.all	colorscheme for all points
cols.g	colorscheme for the points in the highlighted group
...	more plotting parameters

Value

Nothing particularly interesting

setGenomeProps *setGenomeProps*

Description

Set the genome properties for a GRanges or GAlignments object given the name of a genome assembly

Usage

```
setGenomeProps(
  gr,
  assembly,
  dropUnknownChrs = TRUE,
  adjChrNames = TRUE,
  silent = FALSE,
  ...
)
```

Arguments

gr	GRanges object or GAlignments object to modify
assembly	assembly
dropUnknownChrs	discard entries with seqnames not supported by assembly
adjChrNames	should the prefix "chr" be added to main chromosomes if not already present and chrMT be renamed to chrM?
silent	Limit logging to most important messages
...	arguments passed on to `getSeqlengths4assembly`

Value

GRanges object with genome properties set

sortGr *sortGr*

Description

sort a `GRanges` object

Usage

```
sortGr(gr, alnum = FALSE)
```

Arguments

gr	GRanges object to sort
alnum	sort chromosomes alphanumerically instead of by number

Value

sorted GRanges object

---

strTrim *strTrim*

Description

trim a character vector to have a desired length by taking the beginning of the string and the end of the string

Usage

```
strTrim(
  ss,
  len.out = 50,
  trim.str = "...",
  len.pref = ceiling((len.out - nchar(trim.str))/2),
  len.suf = len.out - len.pref - nchar(trim.str)
)
```

Arguments

ss	character vector
len.out	target output length to trim to
trim.str	string to place in between prefix and suffix
len.pref	length of the prefix to be used from the original string
len.suf	length of the suffix to be used from the original string

Value

character vector in which each element has length<=len.out

Author(s)

Fabian Mueller

summarizeSetOverlap *summarizeSetOverlap*

Description

prints overlap statistics for two sets

Usage

```
summarizeSetOverlap(
  set1,
  set2,
  set1name = "set1",
  set2name = "set2",
  doVenn = TRUE
)
```

Arguments

set1	vector containing elements in set 1
set2	vector containing elements in set 2
set1name	name for set 1
set2name	name for set 2
doVenn	plot a Venn diagram

Value

nothing of interest. If doVenn, a venn diagram will be plotted to the current plotting device

Author(s)

Fabian Mueller

Examples

```
summarizeSetOverlap(1:50, 23:100, "1:50", "23:100")
```

testAssoc *testAssoc*

Description

Tests for association between two vectors. Based on RnBeads:::test.traits

Usage

```
testAssoc(x, y, permMat = NULL)
```

Arguments

x	Sample values for the first trait. This must be a vector of type factor, integer or numeric.
y	Sample values for the second trait. This must be a vector of type factor, integer or numeric.
permMat	Matrix of sample permutations (indices of x that will be used in permutation tests) in case none of the traits is a factor, and thus permutation-based p-value from correlations is computed. If this parameter is NULL and both x and y are sequences of numbers, no p-value is calculated.

Value

List of four elements:

error Error, if any, that prevented this function from computing a p-value for trait association.

test Type of test performed. This is one of "Fisher", "Wilcoxon", "Kruskal-Wallis", "Correlation" or NA. The last value indicates that the traits cannot be tested for association.

correlation Value of the pearson correlation coefficient between x and y, or NA if any of them is factor.

pvalue Calculated p-value, or NA if the traits cannot be tested for association.

textSearch *textSearch*

Description

Shortcut wrapper around aggregate to search case insensive in a vector of strings

Usage

```
textSearch(s, x, ...)
```

Arguments

s	string or expression
x	string vector to search in

Value

string vector of matches

Author(s)

Fabian Mueller

---

theme_nogrid *theme_nogrid*

Description

A ggplot2 theme based on theme_bw but with no grid lines and axis only on top and bottom

Usage

```
theme_nogrid(base_size = 8, base_family = "Helvetica")
```

Arguments

base_size	base size
base_family	base family

Value

the theme structure

Examples

```
theme_set(theme_nogrid())
dframe <- data.frame(x=runif(100),y=runif(100))
ggplot(dframe,aes(x=x,y=y)) + geom_point()
```

umapParamGridReport *umapParamGridReport*

Description

Generate a report with plots of UMAP dimension reduction plots for parameter combinations

Usage

```
umapParamGridReport(
  X,
  outDir,
  metric = c("euclidean"),
  min_dist = c(0.01, 0.05, seq(0.1, 0.9, by = 0.1)),
  n_neighbors = c(5, 15, 25, 50),
  ...
)
```

Arguments

X	feature matrix containing one row for each observation and one column for each feature
outDir	output directory
metric	parameters passed on to uwot::umap()
min_dist	parameters passed on to uwot::umap()
n_neighbors	parameters passed on to uwot::umap()
...	parameters passed on to getDimRedPlot

Value

a muReportR report (HTML) showing dimension reduction plots for the grid search

Author(s)

Fabian Mueller

underscore2camel *underscore2camel*

Description

converts underscores to camel case in a (vector of) string(s)

Usage

```
underscore2camel(x)
```

Arguments

a	string or string vector

Value

the converted string(s)

Examples

```
underscore2camel("bla_blubb")
```

unloadPackage *unloadPackage*

Description

unloads a package without quitting the R session. Useful for developing packages interactively

Usage

```
unloadPackage(package.name)
```

Arguments

package.name	name (string) of the package to be reloaded

Value

result of `library(package.name)`

Examples

```
library(GenomicRanges)
unloadPackage("GenomicRanges")
```

writeTab *writeTab*

Description

Wrapper around `write.table` to write tab-separated tables by default

Usage

```
writeTab(
  x,
  fn,
  sep = "\t",
  row.names = FALSE,
  col.names = TRUE,
  quote = FALSE,
  ...
)
```

Arguments

`x`	table to write to file
`fn`	filename to write to
`sep`	see `?write.table`
`row.names`	see `?write.table`
`col.names`	see `?write.table`
`quote`	see `?write.table`
`...`	passed to `write.table`
`stringsAsFactors`	see `?write.table`

Value

the result of `write.table`

Author(s)

Fabian Mueller

Index