Package 'momr'

October 13, 2022

Title Mining Metaomics Data (MetaOMineR)

Version 1.1

Date 2015-07-24

Author Edi Prifti, Emmanuelle Le Chatelier

Maintainer Edi Prifti <edi.prifti@gmail.com>

Description 'MetaOMineR' suite is a set of R packages that offers many functions and modules needed for the analyses
of quantitative metagenomics data. 'momr' is the core package and contains routines for biomarker identification and exploration.
Developed since the beginning of field, 'momr' has evolved and is structured around the different modules
such as preprocessing, analysis, vizualisation, etc. See package help for more information.

License Artistic-2.0

Depends R (>= 2.10)

Imports Hmisc, gplots, nortest

Collate 'normalization.R' 'downsizing.R' 'querries.R' 'filtering.R' 'visualization.R' 'analyses.R' 'data.R' 'mapreduce.R'

LazyLoad yes

License_restricts_use yes

NeedsCompilation no

Repository CRAN

Date/Publication 2015-07-27 12:42:14

R topics documented:

momr-package *Mining Metaomics Data In R*

Description

momr also known as MetaOMineR is a R package that offers many functions and modules needed for the analyses of quantitative metagenomics data. It is conceived for the analyses of whole NGS data but can be used for 16S datasets as well or other type of omics data. Developed since the very beginning of the field the package has evolved and is structured around different modules such as preprocessing, analysis, visualization, etc. This package contains the different algorithms and routines as well as some test data objects. It is used along with other data packages that contain the needed information to describe a given catalogue developped in the same series.

Details

Package:	momr
Type:	Package
Version:	1.1
Date:	2015-07-24
License:	Artistic-2.0

The starting point of the analyses starts with a read count matrix that has been mapped onto a gene catalog. This raw read count matrix can be preprocessed through downsizing, normalization and filtering steps to obtain the abundance frequencies. The samples can then be clustered in different ways to check for similarity and outliers. Genes can than be statistically related to a given phenotype in order to select those that are of most interest (the biomarkers). Genes of interest can be projected onto the MGS catalog to obtain a reduced dataset of microbial entities that is to be further annotated.

Updates

2015/06/24:	First version for CRAN. One year long changes implemented, map reduce, normalization etc...
2014/03/24:	First official release, licence added, and map-reduce procedures
2014/02/03:	added new functions phenoPairwiseRelations, extractSignificant and lmp
2014/02/03:	testRelations modified to give the direction of a correlation
2014/02/03:	MGS sample catalog udate. This version has genes sorted based on the whole metahit cohort

Author(s)

Authors:	Edi Prifti and Emmanuelle Le Chatelier
Maintainer:	Edi Prifti <edi.prifti [at] gmail.com>

References

Le Chatelier, Emmanuelle, Trine Nielsen, Junjie Qin, Edi Prifti, Falk Hildebrand, Gwen Falony, Mathieu Almeida, et al. "Richness of Human Gut Microbiome Correlates with Metabolic Markers." Nature 500, no. 7464: 541???546.

Examples

```
# load the package
library(momr)

#' all the data in the package
# data(package="momr")

#' load the raw and frequency test dataset
data("hs_3.3_metahit_sample_dat_raw")
data("hs_3.3_metahit_sample_dat_freq")


#' NORMALIZATION
#' This should be performed with the whole dataset (complete catalogue).
#' But here is an exemple with the subset of the data for illustration purposes
data(hs_3.3_metahit_genesize)
norm.data <- normFreqRPKM(dat=hs_3.3_metahit_sample_dat_raw, cat=hs_3.3_metahit_genesize)


#' CLUSTERING OF SAMPLES
hc.data <- hierClust(data=hs_3.3_metahit_sample_dat_freq[,1:5], side="col")
clust.order <- hc.data$mat.hclust$order
#' order samples followin the hierarchical clustering
ordered.samples <- colnames(hs_3.3_metahit_sample_dat_freq[,1:5])[clust.order]
#' how close are the two first samples (spearman, rho)
hc.data$mat.rho[ordered.samples[1], ordered.samples[2]]
# select the samples closely related together
close.samples <- filt.hierClust(hc.data$mat.rho, hclust.method = "ward", plot = TRUE, filt = 0.5)

#' CLUSTER GENES ON THE MGS CATALOG
#' load the curated mgs data for the hs_3.3_metahit catalog
data("mgs_hs_3.3_metahit_sup500")

#' project a list of genes onto the mgs
genebag <- rownames(hs_3.3_metahit_sample_dat_freq)
mgs <- projectOntoMGS(genebag=genebag, list.mgs=mgs_hs_3.3_metahit_sup500)

#' extract the profile of a list of genes from the whole dataset
mgs.dat <- extractProfiles(mgs, hs_3.3_metahit_sample_dat_freq, silent=FALSE)

#' plot the barcodes
par(mfrow=c(5,1), mar=c(1,0,0,0))
for(i in 1:5){
  plotBarcode(mgs.dat[[i]])
```

```
}

#' compute the filtered vectors
mgs.mean.vect <- computeFilteredVectors(profile=mgs.dat, type="mean")


#' TEST RELATIONS
#' for the first 1000 genes
res.test <- testRelations(data=hs_3.3_metahit_sample_dat_freq[1:500,],
                          trait=c(rep(1,150),rep(2,142)),type="wilcoxon")
head(res.test)
print(paste("There are",sum(res.test$p<0.05, na.rm=TRUE),"significant genes and",
            sum(res.test$q<0.05, na.rm=TRUE), "after adjustment for multiple testing"))

res.test.mgs <- testRelations(data=mgs.mean.vect,trait=c(rep(1,150),rep(2,142)),type="wilcoxon")


#' DOWNSIZING
#' downsize the matrix
data.downsized <- downsizeMatrix(data=hs_3.3_metahit_sample_dat_raw[,1:5],level=600,repetitions=1)
colSums(data.downsized, na.rm=TRUE)

#' downsize the genecount
data.genenb <- downsizeGC(data=hs_3.3_metahit_sample_dat_raw[,1:5], level=600, repetitions=3)
par(mfrow=c(1,1), mar=c(4,4,4,4))
plot(density(colMeans(data.genenb, na.rm=TRUE)), main="density of downsized gene richness")

#' End of test file
```

---

abondanceScore *abondanceScore*

Description

Computes the sum of the vectors divided by the prevalence

Usage

```
abondanceScore(vect, th = 0)
```

Arguments

vect	: a numerical vector
th	: the threshold to be applied, default is 0

Details

abondanceScore

Value

a mean abundance

Author(s)

Edi Prifti

`aggregateProfiles` *aggregateProfiles*

Description

This function takes a list of profile matrixes and returns an aggregated big matrix. The individual matrixes can be filtered in size so that the first X rows are used for each of them. This function is used to prepare the data and plot different MGS as barcodes.

Usage

```
aggregateProfiles(list.profiles, max.size = 25, min.size = max.size)
```

Arguments

`list.profiles`	: list of matrix profiles
`max.size`	: the maximum number of rows to be selected in the final agregated matrix. default max.size=25.
`min.size`	: this is the minimum number of rows rows to be selected in the final aggregated matrix. If a group has less it will be discarded. By default min.size=max.size.

Details

`aggregateProfiles`

Value

an aggregated profile matrix.

Author(s)

Emmanuelle Le Chatelier

buildMgsFinal *buildMgsFinal @date December 20th 2013*

Description

This function will take a vector of genes (to be transformed into a list of genebags) or a list of genebags and will extract the profiles. Next genes well be ordered by connectivity which is to be computed for each group and the 50 most connected are selected to consitute the marker genes. These will be then used to compute the mean vectors. A final object containing all this information along with taxonomical annotation will be returned

Usage

```
buildMgsFinal(genebag = NULL, mgs.cat, mgs.taxo, profiles, conn = TRUE,
  silent = TRUE, filt = 20)
```

Arguments

genebag	: a vector of genes to be projected onto mgs or a list of genebags, default = NULL.
mgs.cat	: MGS catalogue to be used
mgs.taxo	: taxonomy table for the MGS catalogue
profiles	: the data profile matrix to extract the profiles
conn	: if TRUE the connectivity of a group is to be computed and ordered, default = TRUE.
silent	: print detailled information on progress, default = FALSE.
filt	: filtering based on percentage of prevalence to avoid noise for no signal samples by computeFilteredVectors.

Details

buildMgsFinal

Value

a list containing the final elements such as the 50 most connected genes, the mean vectors etc

Author(s)

Edi Prifti

computeFilteredVectors *computeFilteredVectors*

Description

filters and computes verctors based on gene profiles from a single matrix or a list of matrix profiles

Usage

```
computeFilteredVectors(profile, type = "mean", filt = 0, debug = FALSE)
```

Arguments

profile	: list of (or unique) matrix profiles
type	: vectorisation method, vectors are calculated from the mean, median or sum of a given list of genes; default type="mean" otherwise it will be "median" or "sum"
filt	: filtering threshold in and sparse values put to 0 default filt= 0, no filtering
debug	: default is FALSE, when TRUE information on advancement is printed

Details

computeFilteredVectors

Value

a filtered vector or a matrix of filtered vectors

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

computeSignalMetrics *computeSignalMetrics*

Description

Computes scores of data variation within a given MGS cluster.

Usage

```
computeSignalMetrics(dat)
```

Arguments

dat	: a matrix where operations will be performed on the columns. Please transpose if operations are needed in the rows.

Details

computeSignalMetrics

Value

a matrix of results where scores are in the columns

Author(s)

Edi Prifti

---

computeUpsizedGC *computeUpsizedGC*

Description

This procedure takes a table of meaned downsized gene counts where at least one column is donwsized at a common minimal level. It uses this information to fit distributions of correlations between different downsized levels and "predict" values for the samples that have not the needed sequencing depth. The fitting of the models is based on the n-1 to be closer to reality and avoid accumulating errors.

Usage

```
computeUpsizedGC(richness.table, side = 2, keep.real = TRUE, plot = FALSE)
```

Arguments

richness.table	: matrix with samples in rows and downsizings in the columns as produced by downsizedRichnessL2T
side	: by default is 2 (downsizings in the columns)
keep.real	: by default is TRUE. Substitue the predicted values by the real values when is not NA
plot	: default FALSE, plots the regressions

Details

computeUpsizedGC

Value

matrix with the same dimensions as richness.table but with complete values.

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

See Also

downsizedRichnessL2T and downsizeGC.all

`connectivity` *connectivity*

Description

This function computes the intra-row correlation and applied a threshold to compute the connections of each row. A connectivity vector is returned.

Usage

```
connectivity(prof, method = "pearson", th = 0, soft = FALSE)
```

Arguments

`prof`	: a profile matrix. This data is used to compute correlations.
`method`	: the correlation method, default = pearson.
`th`	: default 0, if >0 than this threshold will be applied to compute a hard threholded connectivity.
`soft`	: default = FALSE, if TRUE and when th > 0, the connectivity is computed as the soft threholded, sum of correlations above the threshold

Details

`connectivity`

Value

a vector of connectivity

Note

this connectivity does not use a hard thresholding but is based on a total correlation score

Author(s)

Emmanuelle le Chatelier & Edi Prifti

countSampledGenes *countSampledGenes*

Description

counts the number of genes that have been sampled by un_indexing

Usage

```
countSampledGenes(v.samp)
```

Arguments

v.samp	: a character vector of the sampled indexed reads (output of sampleRandlomly)

Details

countSampledGenes

Value

a table with counts for each gene

Author(s)

Edi Prifti

countSampledGenesGC *countSampledGenesGC*

Description

counts the number of genes that have been sampled by un_indexing

Usage

```
countSampledGenesGC(v.samp)
```

Arguments

v.samp	: a character vector of the sampled indexed reads (output of sampleRandlomly)

Details

countSampledGenesGC

Value

a table with counts for each gene

Note

an optimized version for the gene count downsizing

Author(s)

Edi Prifti

`deleteData` *deleteData*

Description

This function will delete the original and treated split temporary files to celan the workspace.

Usage

```
deleteData(name = "data_part")
```

Arguments

`name`	: the name of the split files in the disk preceding the incremental number, default="data_part"

Details

`deleteData`

Value

nothing to be returned

Author(s)

Edi Prifti

`downsizedRichnessL2T` *downsizedRichnessL2T*

Description

This procedure takes a list that is the result of the downsizeGC.all method and transforms it in a matrix of meaned downsied values. Each element of this list contains downsizing results for a given sample. This result is a matrix in lines the number of simulations and in columns the different downsizing levels

Usage

```
downsizedRichnessL2T(richness.list)
```

Arguments

`richness.list` : the result of the downsizeGC.all method

Details

`downsizedRichnessL2T`

Value

A matrix with the samples in rows and the downsizing in columns

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

`downsizeGC` *downsizeGC*

Description

This function takes a matrix with raw reads counts and computes the number of genes at a given downsizing level a given number of times.

Usage

```
downsizeGC(data, level = 1.1e+07, repetitions = 30, silent = FALSE)
```

Arguments

data	: raw read count matrix with gene_ids as rownames
level	: default 11e6, the downsizing level number of reads to be selected randomly.
repetitions	: default 30, the number of times the drawing is performed. Usually 30 or 10 to speed things out
silent	: default is FALSE prints the status of downsizing

Details

downsizeGC

Value

a matrix containing in rows a vector for each repetition and in columns the number of downsized genes for each sample

Note

if the downsizing level is higher than the number of reads for a given sample than the result will be NA

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

---

downsizeGC.all *downsizeGC.all*

Description

This function takes a matrix with raw reads counts and computes the number of genes at different downsizing levels a given number of times. This is similar to the downsizeGC function but for optimization purposes it downsizes at different thresholds all together

Usage

```
downsizeGC.all(data, levels = c(seq(1e+06, 1.1e+07, 1e+06)),
  repetitions = 10, silent = FALSE)
```

Arguments

data	: raw read count matrix with gene_ids as rownames
levels	: default seq(1E06,11E06,1E06), the downsizing levels number of reads to be selected randomly.
repetitions	: default 10, the number of times the drawing is performed. Usually 30 or 10 to speed things out
silent	: default is FALSE prints the status of downsizing

Details

downsizeGC.all

Value

a list of matrixes one per sample containing in rows a vector for each repetition and in columns the number of downsized genes for each downsizing level

Note

if the downsizing level is higher than the number of reads for a given sample than the result will be NA

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

---

downsizeMatrix *downsizeMatrix*

Description

takes a matrix with raw read gene counts and converts it to a downsized matrix with identical number of mapped reads for each sample (column)

Usage

```
downsizeMatrix(data, level = 1.1e+07, repetitions = 1, silent = FALSE)
```

Arguments

data	: raw read count matrix with gene_ids as rownames
level	: default 11E06, the downsizing levels number of reads to be selected randomly.
repetitions	: default 1, This can be also computed several times, but one is the error minimal downsizing strategy
silent	: default is FALSE prints the status of downsizing

Details

downsizeMatrix

Value

downsized read gene count matrix corresponding to the mean counts obtained with the selected number of independant downsizing procedure

Note

if the downsizing level is higher than the number of reads for a given sample than the result will be NA

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

`extractProfiles` *extractProfiles*

Description

This function extracts the profiles from a gene profile matrix of a group of genes or a list of groups of genes. It can also restrict the size of the result

Usage

```
extractProfiles(genebag, data, size.max = 15000, size.min = 1,
  silent = TRUE)
```

Arguments

`genebag`	: vector or list of gene_ids
`data`	: raw count or frequency matrix with genes_ids as rawnames
`size.max`	: default 15000, maximum number of profiles to be extracted
`size.min`	: default 1, the minimal size threshold above which a group of genes is selected. This is used to extract multiple profiles and filtering the list with a minimal number of genes
`silent`	: default TRUE, detailling and following computation progress

Details

`extractProfiles`

Value

a matrix or a list of profile matrixes

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

`extractSignificant` *extractSignificant*

Description

This function will extract a list of vectors p- or q-values from an object produced by phenoPairwiseRelations.

Usage

```
extractSignificant(relation.matrix, interest, threshold = 0.05)
```

Arguments

`relation.matrix`	: a matrix of p- produced by or q-values by phenoPairwiseRelations()
`interest`	: a vector of variable names of interest.
`threshold`	: default 0.05 needed to select significant relations

Details

`extractSignificant`

Value

a list of vectors containing p-values or q-values along with the names of the variables.

Author(s)

Edi Prifti

`filt.hierClust` *filt.hierClust*

Description

This function takes as input a square similarity matrix and searches for clusters of samples with strong associations and extracts the sub matrix with the closely related sampless. Only positive correlations are considered here.

Usage

```
filt.hierClust(mat.rho, hclust.method = "ward", side.col.c = NULL,
  side.col.r = NULL, size = 10, plot = TRUE, filt = 0.5)
```

Arguments

mat.rho	: square correlation matrix with ids (can be used for also other than just samples)
hclust.method	: the hierarchical clustering method, by default it is the ward method
side.col.c	: a vector of colors to be applied in the columns, usually depincting a class
side.col.r	: a vector of colors to be applied in the rows, usually depincting a class
size	: the number of samples in the resulting ordered matrix
plot	: logical default TRUE. It will plot the heatmap of the similarity with the hierchical clustering
filt	: default is 0.5 and is the filtering threshold to be applied

Details

filt.hierClust

Value

it will return a matrix with samples in rows and their closely related ones on the columns along with the correlation score.

Author(s)

Emmanuelle Le Chatelier & Edi Prifti

---

filterListGenebags *filterListGenebags*

Description

This function filters a list of mgs with gene_id >= a given mgs gene number.

Usage

```
filterListGenebags(list.genebags, size.min = 0, size.max = 15000)
```

Arguments

list.genebags	: a list of genebags that needs to be filtered
size.min	: the minimal size threshold above which mgs are selected, default 0
size.max	: the maximal size threshold above which mgs are selected, default 15000

Details

filterListGenebags

Value

a list of selected genebags with their original elements (usually geneids)

Note

This is the former filterListMGS function

Author(s)

Emmanuelle Le Chatelier

`filterMat` *filterMat*

Description

This function filters a matrix (mat) based on the rate of positive value (filt) in each individual under a given degree of presence (filt), all sparse values are put to 0.

Usage

```
filterMat(mat, filt = 0)
```

Arguments

`mat`	: matrix of counts
`filt`	: filtering threshold in percentage

Details

`filterMat`

Value

a cleaned matrix

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

hierClust *hierClust*

Description

This function computes the pairwise distance between samples and computes a hierarchical clustering that is further depicted as a heatmap graphic

Usage

```
hierClust(data, side = "col", dist = "correlation", cor.type = "spearman",
  hclust.method = "ward", side.col.c = NULL, side.col.r = NULL,
  plot = TRUE)
```

Arguments

data	: frequency matrix with gene_ids in the rownames
side	: the distance can be performed on the columns or on the rows
dist	: the type of distance used. By default this is correlation based similarity
cor.type	: when correlation matrix, the default is spearman
hclust.method	: the hierarchical clustering method, by default it is the ward method
side.col.c	: a vector of colors to be applied in the columns, usually depincting a class
side.col.r	: a vector of colors to be applied in the rows, usually depincting a class
plot	: logical default TRUE. It will plot the heatmap of the similarity with the hierchical clustering

Details

hierClust

Value

it will return a list of three variables, the correlation matrix, the distance matrix and the hclust object

Note

updated hierClust functions by elechat april 7th 2015 added options SideColors added + spearman == pearson(rank)

Author(s)

Edi Prifti

hs_3.3_metahit_genesize *hs_3.3_metahit_genesize*

Description

The gene size for the hs_3.3_metahit catalogue, needed for the normalization procedure

Details

CATALOG HS_3.3_METAHIT

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

hs_3.3_metahit_sample_dat_freq *hs_3.3_metahit_sample_dat_freq*

Description

this dataset contains only the first 100000 genes of the catalogue for size purposes

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

hs_3.3_metahit_sample_dat_raw *hs_3.3_metahit_sample_dat_raw*

Description

this dataset contains only the first 100000 genes of the catalogue for size purposes

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

`indexReads` *indexReads*

Description

for a given column of the raw count matrix this function indexes the count vector in a vector of reads of the length of the sum of the counts. It allows after to randomly select a subset

Usage

```
indexReads(v, silent = FALSE)
```

Arguments

`v`	: an integer raw count vector
`silent`	: default FALSE, debugging

Details

`indexReads`

Value

an character indexed vector

Author(s)

Edi Prifti

`indexReadsGC` *indexReadsGC*

Description

for a given column of the raw count matrix this function indexes the count vector in a vector of reads of the length of the sum of the counts. It allows after to randomly select a subset

Usage

```
indexReadsGC(v, silent = TRUE)
```

Arguments

`v`	: an integer raw count vector
`silent`	: default FALSE, debugging

Details

indexReadsGC

Value

an character indexed vector

Note

this function is optimized when dowsizing for gene count

Author(s)

Edi Prifti

---

launchTask *launchTask*

Description

This function distributes the calculations as separate processes in a multi-thread server.

Usage

```
launchTask(input, output, script)
```

Arguments

input	: a folder containing elements that are to be processed
output	: a folder where the processed results will be writen
script	: the R script that is to be executed in parallel

Details

launchTask

Value

nothing

Note

the number of processors may be specified. TODO: add the argument

Author(s)

Edi Prifti

lmp *lmp*

Description

This function will extract the p-value from a linear model object. It is used by phenoPairwiseRelations

Usage

```
lmp(modelobject)
```

Arguments

modelobject	: a linear model object as produced by lm()

Details

lmp

Value

a p-value

Author(s)

Edi Prifti

mapResults *mapResults*

Description

This function loads the results once finished and merges them together in one dataframe

Usage

```
mapResults(folder = ".", pattern = "_result.rda", type = "rows")
```

Arguments

folder	: the folder where the results are found
pattern	: the pattern of the split files in the disk preceding the incremental number, default="data_part"
type	: a string c(rows, cols, list) indicating how to merge the data. This should be the same as the one used in the splitData procedure.

Details

mapResults

Value

a merged dataframe : Caution ! The merged results may be shuffeled. It is up to the user to reorder the data accordingly.

Author(s)

Edi Prifti

---

mgs_hs_3.3_metahit_sup500 *mgs_hs_3.3_metahit_sup500*

Description

This is the mgs catalogue of the 3.3 metahit gene catalogue and contains 778 MGS of size greater than 500 genes. It has also been manually curated.

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

---

normFreqRPKM *normFreqRPKM*

Description

converts a raw count matrix onto a frequency matrix using the RPKM normalization method. This method consists of two consecutive steps, first dividing the raw counts by the length of the gene sequence and the second shrinking the signal per column to a sum of 1

Usage

```
normFreqRPKM(dat, cat = NULL)
```

Arguments

dat	: raw counts data matrix with gene_ids as rownames
cat	: the current working catalogue where the reads are mapped and counted, (i.e. hs_3.3_metahit, hs_3.9_metahit) This can also be a vector of genelength values that correspond to the number of rows in the dat matrix and are ordered respectively

Details

normFreqRPKM

Value

a normalized frequency matrix

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

---

normFreqTC *normFreqTC*

Description

converts a raw count matrix onto a frequency matrix using the TC (total count) normalization method. This method consists of scaling the signal by the total counts per each sample

Usage

```
normFreqTC(dat)
```

Arguments

dat	: raw counts data matrix with gene_ids as rownames

Details

normFreqTC

Value

a normalized frequency matrix

Author(s)

Edi Prifti

phenoPairwiseRelations *phenoPairwiseRelations*

Description

This function will compute all the relations between different variables adapting different statistical tests as a function of the data type. It will adjust the p-value matrix for multiple testing.

Usage

```
phenoPairwiseRelations(data, adjust = "BH", verbose = FALSE)
```

Arguments

data	: bioclinical data with variables on the rows and samples on the columns
adjust	: method to adjust for multiple testing (default="BH)
verbose	: default=FALSE. If TRUE information will be printed to follow the progression.

Details

phenoPairwiseRelations

Value

a list of two matrixes containing the p-values and the multiple testing adjustment.

Author(s)

Edi Prifti

plotBarcode *plotBarcode*

Description

plots the intensity of a frequency matrix with a 4-fold color step

Usage

```
plotBarcode(data, main = "")
```

Arguments

data	: a frequency matrix to be visualized
main	: the main title of the plot empty by default

Details

plotBarcode

Value

nothing

Note

this may be slightly affected by the size of the catalogue when comparing different studies

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

---

plotBarcode2 *plotBarcode2*

Description

plots the intensity of a frequency matrix with a 4-fold color step. Usually used for complex figures where different MGS are overlapped and annotated with different data.

Usage

```
plotBarcode2(data, main = "", ylabl = "", ylabr = "",
  col.axisl = "white", col.axisr = "white", box = FALSE)
```

Arguments

data	: a frequency matrix to be visualized
main	: the main title of the plot empty by default
ylabl	: label for the left y axis
ylabr	: label for the right y axis
col.axisl	: color for the left y axis
col.axisr	: color for the right y axis
box	: default FALSE.

Details

plotBarcode2

Value

nothing

Note

this may be slightly affected by the size of the catalogue when comparing different studies

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

`plotBarcodeBW` *plotBarcodeBW*

Description

plots in black when a signal is different from zero and white otherwise

Usage

```
plotBarcodeBW(data, main = "")
```

Arguments

`data`	: a frequency matrix to be visualized
`main`	: the main title of the plot empty by default

Details

`plotBarcodeBW`

Value

nothing

Author(s)

Edi Prifti

plotCors *plotCors*

Description

plots a heatmap of a matrix composed of correlation values from -1 to 1. The blue colors are negative correlations while the red are positive

Usage

```
plotCors(data, main = "")
```

Arguments

data	: a frequency matrix to be visualized
main	: the main title of the plot empty by default

Details

plotCors

Value

nothing

Author(s)

Edi Prifti

plotCors2 *plotCors2*

Description

similar to plotCors but with more levels of colors

Usage

```
plotCors2(data, main = "")
```

Arguments

data	: a frequency matrix to be visualized
main	: the main title of the plot empty by default

Details

plotCors2

Value

nothing

Author(s)

Edi Prifti

---

plotMGSQuality *plotMGSQuality*

Description

Visualized scores of data variation within a given MGS cluster as well as the barcode of the MGS. A subset of 50 most connected genes is also plotted the same way.

Usage

```
plotMGSQuality(dat, main = "mgs", scores = TRUE)
```

Arguments

dat	: a matrix where operations will be performed on the columns. Please transpose if operations are needed in the rows. This is typically an MGS frequency matrix with genes in the rows.
main	: the name of the plot
scores	: weather the function should return the computed scores or not. By default is TRUE.

Details

plotMGSQuality

Value

a matrix of results where scores are in the columns

Author(s)

Edi Prifti

plotPvals *plotPvals*

Description

plots a heatmap of a matrix composed of p-values. Gray is not significant at p=0.05 and significance decrases from skyblue to darkred

Usage

```
plotPvals(data, main = "")
```

Arguments

data	: a frequency matrix to be visualized
main	: the main title of the plot empty by default

Details

plotPvals

Value

nothing

Author(s)

Edi Prifti

presenceScore *presenceScore*

Description

Computes the percentage [0,1]of values of a vector that are aboove a given threshold

Usage

```
presenceScore(vect, th = 0)
```

Arguments

vect	: a numerical vector
th	: the threshold to be applied, default is 0

Details

presenceScore

Value

percentage

Author(s)

Edi Prifti

---

profiles2Genebags *profiles2Genebags*

Description

This function will extract from a list of group profiles a list of identifiers of the matrix.

Usage

```
profiles2Genebags(profiles)
```

Arguments

profiles : a list of dataframes

Details

profiles2Genebags

Value

a list of genebags

Author(s)

Edi Prifti

projectOntoMGS *projectOntoMGS*

Description

This function takes a list of genes and projects it to a mgs catalogue

Usage

```
projectOntoMGS(genebag, list.mgs, res.filt.mode = "size",
  res.filt.threshold = 50, not_projected = TRUE)
```

Arguments

genebag	: vector of gene_ids
list.mgs	: this is the structured MGS information formatted as a list of geneids each corresponding to an MGS
res.filt.mode	: the filtering method, either 'perc' or 'size', default will be size. perc : genes projected onto mgs are only kept if they exceed a threshold percentage of the mgs size size : genes projected onto mgs are only kept if they exceed a threshold number of genes
res.filt.threshold	: the threshold used to retain an mgs according to the filtering mode, either 'perc' or 'size' if 'perc' : the minimal percentage of genes projected onto mgs (number of genes projected / mgs size) needed to select a mgs if 'size' : the minimal number of genes projected onto mgs
not_projected	: whether a last genebag containing not projected genes is to be added to the list of selected mgs

Details

projectOntoMGS

Value

a list of selected mgs with the geneids according to the projected genes

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

`sampleRandomly` *sampleRandomly*

Description

This function samples randomly a unique subset of the given indexed vector

Usage

```
sampleRandomly(v.ind, level = 1.1e+07)
```

Arguments

`v.ind`	: a character vector of the indexed reads
`level`	: default 11e6, the number of reads to be selected randomly. This sould be smaller than the size of v.ind

Details

`sampleRandomly`

Value

a character indexed vector

Author(s)

Edi Prifti

`selectListSize` *selectListSize*

Description

This function will sextract a part of a list based on the length of its components. Typically a genebag list can be used. This function will work for uniclass lists of vectors and data frames.

Usage

```
selectListSize(l, min.size = 0, max.size = 0, names = FALSE)
```

Arguments

`l`	: a list of vectors or matrixes.
`min.size`	: default = 0 and this has no action. This is the lower threshold for the element's size.
`max.size`	: default = 0 and this has no action. This is the upper threshold for the element's size.
`names`	: default = FALSE the function will return a subset of the list, if TRUE the function will return only the names of the elements of the list as identifiers.

Details

`selectListSize`

Value

the trimmed subset of the list.

Author(s)

Edi Prifti

`splitData` *splitData*

Description

This function splits a dataframe object on a given number of equally sized shares and saves them in the disk as RData objects with an incremental name.

Usage

```
splitData(data, shares = 30, name = "data_part", rows = TRUE)
```

Arguments

`data`	: the dataframe to be split
`shares`	: the number of shares, default=30
`name`	: the name of the split files in the disk preceding the incremental number, default="data_part"
`rows`	: logical parameter, if default rows=="TRUE" the rows will be split

Details

`splitData`

Value

does not return anything

Author(s)

Edi Prifti

`testRelations` *testRelations*

Description

This function applies a statistical test either a correlation (spearman, pearson), wilcoxon or t.test as a function of a given phenotype. It will return a matrix of probabilities p and q values along with the correlation coefficient or the enrichment variable when a binary parameter.

Usage

```
testRelations(data, trait, type, restrict = rep(TRUE, ncol(data)),
  multiple.adjust = "BH", paired = FALSE, debug = FALSE)
```

Arguments

`data`	: frequency matrix with gene_ids in the rownames
`trait`	: a vector with the trait to test, binary or numerical variable
`type`	: a character string indicating the type of test to be applied
`restrict`	: an optional logical vector to select a subset of the samples to perform the test default restrict = rep(TRUE, ncol(data)) ie all the samples are selected
`multiple.adjust`	: type of multiple adjustment default is "BH" i.e. Benjamini & Hochberg method
`paired`	: logical with default FALSE wether the test should be paired or not
`debug`	: default FALSE, when TRUE the progress is printed each 1000 steps

Details

`testRelations`

Value

a matrix with analytical results (correlation tests) indicating rho, rho2, p and q values for each parameter tested

Note

New addon taking into account a trait for correlation, when it is a two class variable with the same number of elements a correlation between both groups is performed

Author(s)

Edi Prifti & Emmanuelle Le Chatelier

`watchProgress` *watchProgress*

Description

This function prints the progress on the input and output folders by comparing the number of output objects.

Usage

```
watchProgress(input, output)
```

Arguments

`input`	: a folder containing elements that are to be processed
`output`	: a folder where the processed results will be writen

Details

`watchProgress`

Value

nothing

Author(s)

Edi Prifti

Index