Package: cogena (via r-universe)

November 1, 2026

Version 1.46.0

Title co-expressed gene-set enrichment analysis

Description cogena is a workflow for co-expressed gene-set enrichment analysis. It aims to discovery smaller scale, but highly correlated cellular events that may be of great biological relevance. A novel pipeline for drug discovery and drug repositioning based on the cogena workflow is proposed. Particularly, candidate drugs can be predicted based on the gene expression of disease-related data, or other similar drugs can be identified based on the gene expression of drug-related data. Moreover, the drug mode of action can be disclosed by the associated pathway analysis. In summary, cogena is a flexible workflow for various gene set enrichment analysis for co-expressed genes, with a focus on pathway/GO analysis and drug repositioning.

biocViews Clustering, GeneSetEnrichment, GeneExpression, Visualization, Pathways, KEGG, GO, Microarray, Sequencing, SystemsBiology, DataRepresentation, DataImport

Depends R (>= 3.6), cluster, ggplot2, kohonen

Imports methods, class, gplots, mclust, amap, apcluster, foreach, parallel, doParallel, fastcluster, corrplot, biwt, Biobase, reshape2, stringr, tibble, tidyr, dplyr, devtools

Suggests knitr, rmarkdown (>= 2.1)

Collate dist_class.R genecl_class.R cogena_class.R Class_methods.R cogena.R upDownGene.R PEI.R sota.R pClusters.R coExp.R clEnrich.R clEnrich_one.R enrichment.R gene2set.R geneInCluster.R geneExpInCluster.R corInCluster.R gmt2list.R gmtlist2file.R heatmap.3.R heatmapCluster.R heatmapPEI.R heatmapCmap.R

License LGPL-3

LazyData true

Encoding UTF-8

URL https://github.com/zhilongjia/cogena

NeedsCompilation no

BugReports https://github.com/zhilongjia/cogena/issues

VignetteBuilder knitr

RoxygenNote 7.0.2

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev git make libharfbuzz-dev libgit2-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

Repository Bioconductor 3.23

Date/Publication 2026-04-28 12:41:05 UTC

RemoteUrl https://github.com/bioc/cogena

RemoteRef RELEASE_3_23

RemoteSha dc5c79e1d2d28431e42190b3c16e06bc15c7db51

Contents

AllGeneSymbols . . . 3
clEnrich . . . 3
clEnrich_one . . . 4
clusterMethods . . . 6
coExp . . . 7
cogena-class . . . 9
cogena_package . . . 10
corInCluster . . . 11
DEexprs . . . 12
enrichment . . . 13
gene2set . . . 14
genecl-class . . . 15
geneclusters . . . 16
geneExpInCluster . . . 17
geneInCluster . . . 18
gmt2list . . . 19
gmtlist2file . . . 20
heatmapCluster . . . 20
heatmapCmap . . . 22
heatmapPEI . . . 25
mat . . . 27
nClusters . . . 28
PEI . . . 29
Psoriasis . . . 30
sampleLabel . . . 30
show,cogena-method . . . 31
sota . . . 31
summary,genecl-method . . . 34
upDownGene . . . 34

Index **36**

`AllGeneSymbols` *All the gene symbols*

Description

All the gene symbols

Format

a vector with 18986 gene symbols.

Source

http://www.genenames.org/

`clEnrich` *Gene set enrichment for clusters*

Description

Gene set enrichment for clusters sourced from coExp function. the enrichment score are based on -log2(p) with p from hyper-geometric test.

Usage

```
clEnrich(
  genecl_obj,
  annofile = NULL,
  sampleLabel = NULL,
  TermFreq = 0,
  ncore = 1
)
```

Arguments

`genecl_obj`	a genecl object
`annofile`	gene set annotation file
`sampleLabel`	sameple Label. Do make the label of interest located after the control label in the order of factor. See details.
`TermFreq`	a value from [0,1) to filter low-frequence gene sets
`ncore`	the number of cores used

Details

sampleLable: Use factor(c("Normal", "Cancer", "Normal"), levels=c("Normal", "Cancer")), instead of factor(c("Normal", "Cancer","Normal")). This parameter will affect the direction of gene regulation in cogena.

Gene sets availiable (See vignette for more):

- c2.cp.kegg.v7.01.symbols.gmt.xz (From Msigdb)
- c2.cp.reactome.v7.01.symbols.gmt.xz (From Msigdb)
- c5.bp.v7.01.symbols.gmt.xz (From Msigdb)

Value

a list containing the enrichment score for each clustering methods and cluster numbers included in the genecl_obj

Source

Gene sets are from

1. http://www.broadinstitute.org/gsea/msigdb/index.jsp

2. http://amp.pharm.mssm.edu/Enrichr/

Examples

```
#annotaion
annoGMT <- "c2.cp.kegg.v7.01.symbols.gmt.xz"
annofile <- system.file("extdata", annoGMT, package="cogena")

utils::data(Psoriasis)
clMethods <- c("hierarchical","kmeans","diana","fanny","som","model","sota","pam","clara","agnes")
genecl_result <- coExp(DEexprs, nClust=2:3, clMethods=c("hierarchical","kmeans"),
    metric="correlation", method="complete", ncore=2, verbose=TRUE)

clen_res <- clEnrich(genecl_result, annofile=annofile, sampleLabel=sampleLabel)
```

---

clEnrich_one *Gene set enrichment for clusters (for one clustering method and a certain number of clusters)*

Description

Gene set enrichment for clusters sourced from coExp function. the enrichment score are based on -log(p) with p from hyper-geometric test.

Usage

```
clEnrich_one(
  genecl_obj,
  method,
  nCluster,
  annofile = NULL,
  sampleLabel = NULL,
  TermFreq = 0
)
```

Arguments

genecl_obj	a genecl or cogena object
method	as clMethods in genecl function
nCluster	as nClust in cogena function
annofile	gene set annotation file
sampleLabel	sameple Label. Do make the label of interest located after the control label in the order of factor. See details.
TermFreq	a value from [0,1) to filter low-frequence gene sets

Details

Gene sets availiable (See vignette for more):

- c2.cp.kegg.v7.01.symbols.gmt.xz (From Msigdb)
- c2.cp.reactome.v7.01.symbols.gmt.xz (From Msigdb)
- c5.bp.v7.01.symbols.gmt.xz (From Msigdb)

Value

a list containing the enrichment score for each clustering methods and cluster numbers included in the genecl_obj

Source

Gene sets are from

1. http://www.broadinstitute.org/gsea/msigdb/index.jsp
2. http://amp.pharm.mssm.edu/Enrichr/

Examples

```
#annotaion
annoGMT <- "c2.cp.kegg.v7.01.symbols.gmt.xz"
annofile <- system.file("extdata", annoGMT, package="cogena")

data(Psoriasis)
clMethods <- c("hierarchical","kmeans","diana","fanny","som","model","sota","pam","clara","agnes")
```

```
genecl_result <- coExp(DEexprs, nClust=2:3, clMethods=c("hierarchical","kmeans"),
    metric="correlation", method="complete", ncore=2, verbose=TRUE)

clen_res <- clEnrich_one(genecl_result, "kmeans", "3", annofile=annofile, sampleLabel=sampleLabel)
clen_res1 <- clEnrich_one(clen_res, "hierarchical", "2", annofile=annofile, sampleLabel=sampleLabel)
```

`clusterMethods` *Basic methods for a genecl object.*

Description

clusterMethods: get the methods of clustering used.

Usage

```
clusterMethods(object)

## S4 method for signature 'genecl'
clusterMethods(object)

## S4 method for signature 'cogena'
clusterMethods(object)
```

Arguments

`object`	a genecl or cogena object

Value

clusterMethods: a character vector.

Examples

```
data(Psoriasis)
genecl_result <- coExp(DEexprs, nClust=2:3,
clMethods=c("hierarchical","kmeans"), metric="correlation",
method="complete",
ncore=1, verbose=TRUE)
clusterMethods(genecl_result)
```

coExp *co-expressed gene-set enrichment analysis*

Description

Co-expressed gene-set enrichment analysis. Gene sets could be Pathway, Gene ontology. The gene co-expression is obtained by various clustering methods.

Usage

```
coExp(
  obj,
  nClust,
  clMethods = "hierarchical",
  metric = "correlation",
  method = "complete",
  ncore = 2,
  verbose = FALSE,
  ...
)
```

Arguments

obj	Differentially expressed gene (DEG) expression profilings. Either a numeric matrix, a data.frame, or an ExpressionSet object. Data frames must contain all numeric columns. In all cases, the rows are the items to be clustered (e.g., genes), and the columns are the samples.
nClust	A numeric vector giving the numbers of clusters to be evaluated. e.g., 2:6 would evaluate the number of clusters ranging from 2 to 6.
clMethods	A character vector giving the clustering methods. The default is "hierarchical". Available options are "hierarchical", "kmeans", "diana", "fanny", "som", "model", "sota", "pam", "clara", "apcluster", and "agnes", with multiple choices allowed.
metric	the distance measure to be used. This should be one of "euclidean", "maximum", "manhattan", "canberra", "binary", "pearson", "abspearson", "correlation", "abscorrelation", "NMI", "biwt", "spearman" or "kendall". Any unambiguous substring can be given. In detail, please reference the parameter method in amap::Dist. Some of the cluster methods could use only part of the metric. See Detail.
method	For hierarchical clustering (hierarchical and agnes), the agglomeration method used. The default is "complete". Available choices are "ward", "single", "complete", and "average".
ncore	Number of core used. The default is 2.
verbose	verbose.
...	to interal function vClusters.

Details

For metric parameter, "hierarchical","kmeans","diana","fanny","pam" and "agnes" can use all the metrics. "clara" uses "manhattan" or "euclidean", other metric will be changed as "euclidean". "sota" uses "correlation" or "euclidean", other metric will be changed as "euclidean". "model" uses its own metric and "som" and "ap" uses euclidean only, which is irrelative with metric.

method: Available distance measures are (written for two vectors x and y):

- euclidean Usual square distance between the two vectors (2 norm).
- maximum Maximum distance between two components of x and y (supremum norm).
- manhattan Absolute distance between the two vectors (1 norm).
- canberra $sum(|x_i - y_i|/|x_i + y_i|)$ Terms with zero numerator and denominator are omitted from the sum and treated as if the values were missing.
- binary (aka asymmetric binary): The vectors are regarded as binary bits, so non-zero elements are 'on' and zero elements are 'off'. The distance is the proportion of bits in which only one is on amongst those in which at least one is on.
- pearson Also named "not centered Pearson" $1 - sum(x_i y_i)/sqrt[sum(x_i^2)sum(y_i^2)]$.
- abspearson Absolute Pearson $1 - |sum(x_i y_i)/sqrt[sum(x_i^2)sum(y_i^2)]|$.
- correlation Also named "Centered Pearson" $1 - corr(x, y)$.
- abscorrelation Absolute correlation $1 - |corr(x, y)|$.
- spearman Compute a distance based on rank.
- kendall Compute a distance based on rank. $\sum_{i,j} K_{i,j}(x, y)$ with $K_{i,j}(x, y)$ is 0 if x_i, x_j in same order as y_i, y_j, 1 if not.
- NMI normalised mutual information. (use correlation instead so far!)
- biwt a weighted correlation based on Tukey's biweight

Value

a genecl object

See Also

clEnrich

Examples

```
data(Psoriasis)

#cogena parameters
# the number of clusters. A vector.
nClust <- 2:6
# the number of cores.
ncore <- 2
# the clustering methods
clMethods <- c("hierarchical","kmeans")
# the distance metric
metric <- "correlation"
```

```
# the agglomeration method used for hierarchical clustering (hierarchical
#and agnes)
method <- "complete"

# See examples of clEnrich function
# the co-expression analysis
## Not run:
genecl_result <- coExp(DEexprs, nClust=nClust, clMethods=clMethods,
metric=metric, method=method, ncore=ncore, verbose=TRUE)

## End(Not run)
```

`cogena-class` *An S4 class to represent co-expressed gene-set enrichment analysis result.*

Description

An S4 class to represent co-expressed gene-set enrichment analysis result.

Slots

`mat` Differentially expressed gene expression profilings. Either a numeric matrix, a data.frame, or an ExpressionSet object. Data frames must contain all numeric columns. In all cases, the rows are the items to be clustered (e.g., genes), and the columns are the samples.

`clusterObjs` a list contains clustering results.

`Distmat` the distance matrix.

`measures` a list of the enrichment results.

`upDn` the enrichment score for up or down-regulated genes.

`clMethods` clustering method.

`labels` the label of genes

`nClust` A numeric vector giving the numbers of clusters to be evaluated. e.g., 2:6 would evaluate the number of clusters ranging from 2 to 6.

`metric` the distance measure to be used. It must be one of "euclidean","maximum", "manhattan", "canberra", "binary", "pearson", "abspearson", "correlation", "abscorrelation", "spearman" or "kendall". Any unambiguous substring can be given. In detail, please reference the parameter method in amap::Dist. Some of the cluster methods could use only part of the metric. Please reference the manual of cogena.

`method` For hierarchical clustering (hclust and agnes), the agglomeration method used. The default is "complete". Available choices are "ward", "single", "complete", and "average".

`annotation` logical matrix of biological annotation with row be DE gene column be gene sets and value be logical.

`sampleLabel` character vector with names are sample names. Only used for plotting.

`ncore` the number of cores used.

`gmt` the gmt file used

`call` the called function

cogena_package *Co-expressed gene set enrichment analysis*

Description

To discovery smaller scale, but highly correlated cellular events that may be of great biological relevance, co-expressed gene set enrichment analysis, cogena, clusters gene expression profiles (coExp) and then make enrichment analysis for each clusters (clEnrich) based on hyper-geometric test. The heatmapCluster and heatmapPEI can visualise the results. See vignette for the detailed workflow.

Source

https://github.com/zhilongjia/cogena

Examples

```
## A quick start

# Loading the examplar dataseat
data(Psoriasis)

# Clustering the gene expression profiling
clMethods <- c("hierarchical","kmeans","diana","fanny","som","model","sota","pam","clara","agnes")
genecl_result <- coExp(DEexprs, nClust=5:6, clMethods=clMethods,
    metric="correlation", method="complete", ncore=2, verbose=TRUE)

# Gene set used
annofile <- system.file("extdata", "c2.cp.kegg.v7.01.symbols.gmt.xz", package="cogena")

# Enrichment analysis for clusters
clen_res <- clEnrich(genecl_result, annofile=annofile, sampleLabel=sampleLabel)

summary(clen_res)


# Visualisation
heatmapCluster(clen_res, "hierarchical", "6")
heatmapPEI(clen_res, "hierarchical", "6", printGS=FALSE)

# Obtain genes in a certain cluster
head(geneInCluster(clen_res, "hierarchical", "6", "2"))

## The end
```

corInCluster *Correlation in the cluster of a cogena object*

Description

Correlation in the cluster of a cogena object. This is helpful if the number of genes in cluster are small.

Usage

```
corInCluster(
  object,
  method,
  nCluster,
  ith,
  corMethod = "pearson",
  plotMethod = "circle",
  type = "upper",
  ...
)

## S4 method for signature 'cogena'
corInCluster(
  object,
  method = clusterMethods(object),
  nCluster = nClusters(object),
  ith,
  corMethod = "pearson",
  plotMethod = "circle",
  type = "upper",
  ...
)
```

Arguments

object	a cogena object
method	a clustering method
nCluster	cluster number
ith	the i-th cluster (should no more than nCluster)
corMethod	a character string indicating which correlation coefficient (or covariance) is to be computed. One of "pearson" (default), "kendall", or "spearman", can be abbreviated.
plotMethod	the visualization method of correlation matrix to be used. Currently, it supports seven methods, named "circle" (default), "square", "ellipse", "number", "pie", "shade" and "color". See examples in corrplot for details

type	"full" (default), "upper" or "lower", display full matrix, lower triangular or upper triangular matrix. See examples in corrplot for details
...	other parameters to corrplot function.

Value

a correlation figure.

See Also

clEnrich corrplot

Examples

```
data(Psoriasis)
annofile <- system.file("extdata", "c2.cp.kegg.v7.01.symbols.gmt.xz",
package="cogena")

## Not run:
genecl_result <- coExp(DEexprs, nClust=2:3, clMethods=c("hierarchical","kmeans"),
    metric="correlation", method="complete", ncore=2, verbose=TRUE)

clen_res <- clEnrich(genecl_result, annofile=annofile, sampleLabel=sampleLabel)

corInCluster(clen_res, "kmeans", "3", "3")
corInCluster(clen_res, "kmeans", "3", "3", plotMethod="square")

## End(Not run)
```

DEexprs *gene expression of DEG*

Description

gene expression of DEG

Format

matrix with 706 DEGs (row) and 116 samples (column).

Source

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE13355

enrichment *get the enrichment table from a cogena object.*

Description

get the enrichment table from a cogena object with certain clustering methods and number of clusters.

Usage

```
enrichment(
  object,
  method,
  nCluster,
  CutoffNumGeneset = Inf,
  CutoffPVal = 0.05,
  orderMethod = "max",
  roundvalue = TRUE,
  add2 = FALSE
)

## S4 method for signature 'cogena'
enrichment(
  object,
  method,
  nCluster,
  CutoffNumGeneset = Inf,
  CutoffPVal = 0.05,
  orderMethod = "max",
  roundvalue = TRUE,
  add2 = TRUE
)
```

Arguments

object	a genecl or cogena object
method	as clMethods in genecl function
nCluster	as nClust in cogena function.
CutoffNumGeneset	the cut-off of the number of gene sets in the return table
CutoffPVal	the cut-off of p-value. The default is 0.05.
orderMethod	the order method, default is max, other options are "mean", "all", "I", "II" or a number meaning the ith cluster.
roundvalue	The default is TRUE. whether or not round the data. such as round(1.54, 1)=1.5
add2	enrichment score for add Up and Down reuglated genes.

Details

orderMethod:

- max. ordered by the max value in clusters beside all
- mean. ordered by the mean value in clusters beside all
- All. ordered by all genes
- I. ordered by the I cluster in two clusters (Up or Down-regulated, add2 should be TRUE)
- II. ordered by the II cluster in two clusters (Up or Down-regulated, add2 should be TRUE)
- a character number. like "3".

Value

a matrix with clusters in row and gene-sets in column.

Examples

```
data(Psoriasis)
annofile <- system.file("extdata", "c2.cp.kegg.v7.01.symbols.gmt.xz",
package="cogena")

## Not run:
genecl_result <- coExp(DEexprs, nClust=2:3, clMethods=c("hierarchical","kmeans"),
    metric="correlation", method="complete", ncore=2, verbose=TRUE)

clen_res <- clEnrich(genecl_result, annofile=annofile, sampleLabel=sampleLabel)

enrichment.table1 <- enrichment(clen_res, "kmeans", "3")
enrichment.table2 <- enrichment(clen_res, "kmeans", "3",
CutoffNumGeneset=10, orderMethod="mean")

## End(Not run)
```

---

gene2set *generate relationship between genes and gene-sets*

Description

Generate relationship between genes (gene SYMBOL) and gene-sets, such as Pathway or GO.

Usage

```
gene2set(annofile = NULL, genenames, TermFreq = 0)

annotationListToMatrix(annotation, genenames)
```

Arguments

annofile	a gmt file. Examples are from MSigDB Collections. A list of gene set could be find in the vignette of cogena
genenames	a SYMBOL gene names charactic vector.
TermFreq	a threshold for the Term Frequence. Default is zero.
annotation	a value returned by gmt2list.

Value

an gene and gene-set relationship matrix

Examples

```
data(Psoriasis)

#annotaion
annoGMT <- "c2.cp.kegg.v7.01.symbols.gmt.xz"
annofile <- system.file("extdata", annoGMT, package="cogena")
# the DEG gene-sets matrix
anno <- gene2set(annofile, rownames(DEexprs))
```

---

genecl-class *An S4 class to represent co-expressed gene*

Description

An S4 class to represent co-expressed gene

Slots

`mat` Differentially expressed gene expression profilings. Either a numeric matrix, a data.frame, or an ExpressionSet object. Data frames must contain all numeric columns. In all cases, the rows are the items to be clustered (e.g., genes), and the columns are the samples.

`clusterObjs` a list contains clustering results.

`Distmat` the distance matrix.

`clMethods` clustering method.

`labels` the label of genes

`nClust` A numeric vector giving the numbers of clusters to be evaluated. e.g., 2:6 would evaluate the number of clusters ranging from 2 to 6.

`metric` the distance measure to be used. It must be one of "euclidean","maximum", "manhattan", "canberra", "binary", "pearson", "abspearson", "correlation", "abscorrelation", "spearman" or "kendall". Any unambiguous substring can be given. In detail, please reference the parameter method in amap::Dist. Some of the cluster methods could use only part of the metric. Please reference the manual of cogena.

`method` For hierarchical clustering (hclust and agnes), the agglomeration method used. The default is "complete". Available choices are "ward", "single", "complete", and "average".

`ncore` the number of cores used.

`call` the called function

`geneclusters` *geneclusters: get the cluster information of a certain clustering method with a certain number.*

Description

geneclusters: get the cluster information of a certain clustering method with a certain number.

Usage

```
geneclusters(object, method, nClust)

## S4 method for signature 'genecl'
geneclusters(object, method = clusterMethods(object), nClust)

## S4 method for signature 'cogena'
geneclusters(object, method = clusterMethods(object), nClust)
```

Arguments

`object`	a genecl or cogena object
`method`	as clMethods in genecl function
`nClust`	cluster numbers

Value

geneclusters: a list or hclust depends on the method

Examples

```
## Not run:
geneclusters(genecl_result, "kmeans", 3)
geneclusters(genecl_result, "hierarchical", 4)

## End(Not run)
```

geneExpInCluster *Get gene names in each clusters and the expression profiling.*

Description

Get gene names in each clusters and the expression profiling. This output is helpful if user want to analyse the data for other application.

Usage

```
geneExpInCluster(object, method, nCluster)

## S4 method for signature 'cogena'
geneExpInCluster(
  object,
  method = clusterMethods(object),
  nCluster = nClusters(object)
)
```

Arguments

object	a genecl or cogena object
method	as clMethods in genecl function
nCluster	as nClust in cogena function.

Value

a list containing a matrix of cluster_id with expression profiling and label a vector of the sample labels.

See Also

clEnrich

Examples

```
data(Psoriasis)
annofile <- system.file("extdata", "c2.cp.kegg.v7.01.symbols.gmt.xz",
package="cogena")

## Not run:
genecl_result <- coExp(DEexprs, nClust=2:3, clMethods=c("hierarchical","kmeans"),
    metric="correlation", method="complete", ncore=2, verbose=TRUE)

clen_res <- clEnrich(genecl_result, annofile=annofile, sampleLabel=sampleLabel)

#geneExpInCluster
geneExp <- geneExpInCluster(clen_res, "kmeans", "3")
```

```
## End(Not run)
```

---

geneInCluster *Get gene names in a certain cluster.*

Description

Get gene names in a certain cluster. This is helpful if user want to get the detail of a cluster.

Usage

```
geneInCluster(object, method, nCluster, ith)

## S4 method for signature 'cogena'
geneInCluster(
  object,
  method = clusterMethods(object),
  nCluster = nClusters(object),
  ith
)
```

Arguments

object	a cogena object
method	a clustering method
nCluster	cluster number
ith	the i-th cluster (should no more than nCluster)

Value

a character vector containing the gene names.

See Also

clEnrich

Examples

```
data(Psoriasis)
annofile <- system.file("extdata", "c2.cp.kegg.v7.01.symbols.gmt.xz",
package="cogena")

## Not run:
genecl_result <- coExp(DEexprs, nClust=2:3, clMethods=c("hierarchical","kmeans"),
    metric="correlation", method="complete", ncore=2, verbose=TRUE)
```

```
clen_res <- clEnrich(genecl_result, annofile=annofile, sampleLabel=sampleLabel)

#summay this cogena object
summary(clen_res)

#geneInCluster
g1 <- geneInCluster(clen_res, "kmeans", "3", "2")

#Up or Down genes with setting nCluster as "2".
g2 <- geneInCluster(clen_res, "kmeans", "2", "1")

## End(Not run)
```

`gmt2list` *read gmt file as a list*

Description

read Gene Matrix Transposed (gmt) file and output a list with the the first column as the names of items in the list. see Gene Matrix Transposed file format for more details.

Usage

```
gmt2list(annofile)
```

Arguments

`annofile`	a gmt file. Examples are from MSigDB Collections. A list of gene set could be find in the vignette of cogena

Value

a gmt list

See Also

gmtlist2file

Examples

```
anno <- "c2.cp.kegg.v7.01.symbols.gmt.xz"
annofile <- system.file("extdata", anno, package="cogena")
gl <- gmt2list(annofile)
```

gmtlist2file *write gmt list into gmt file*

Description

write gmt list into gmt file

Usage

```
gmtlist2file(gmtlist, filename)
```

Arguments

gmtlist	a list containing gmt
filename	output filename

Value

NA

See Also

gmt2list

Examples

```
anno <- "c2.cp.kegg.v7.01.symbols.gmt.xz"
annofile <- system.file("extdata", anno, package="cogena")
gl <- gmt2list(annofile)
gmtfile <- gmtlist2file(gl, filename="")
```

heatmapCluster *heatmap of gene expression profilings with cluster indication.*

Description

heatmap of gene expression profilings with cluster-based color indication. The direction of DEGs are based on latter Vs former from sample labels. For example, labels are as.factor(c("ct", "Disease")), the "Disease" are latter compared with "ct". Usually, the order is the alphabet.

Usage

```
heatmapCluster(
  object,
  method,
  nCluster,
  scale = "row",
  sampleColor = NULL,
  clusterColor = NULL,
  clusterColor2 = NULL,
  heatmapcol = NULL,
  maintitle = NULL,
  printSum = TRUE,
  add2 = TRUE,
  cexCol = NULL,
  ...
)

## S4 method for signature 'cogena'
heatmapCluster(
  object,
  method = clusterMethods(object),
  nCluster = nClusters(object),
  scale = "row",
  sampleColor = NULL,
  clusterColor = NULL,
  clusterColor2 = NULL,
  heatmapcol = NULL,
  maintitle = NULL,
  printSum = TRUE,
  add2 = TRUE,
  cexCol = NULL,
  ...
)
```

Arguments

object	a genecl or cogena object
method	as clMethods in genecl function
nCluster	as nClust in cogena function.
scale	character indicating if the values should be centered and scaled in either the row direction or the column direction, or none. The default is "row".
sampleColor	a color vector with the sample length. The default is from topo.colors randomly.
clusterColor	a color vector with the cluster length. The default is rainbow(nClusters(object)).
clusterColor2	a color vector with 2 elements. The default is rainbow(2).
heatmapcol	col for heatmap. The default is greenred(75).

maintitle	a character. like GSExxx. the output of figure will like "kmeans 3 Clusters GSExxx" in two lines.
printSum	print the summary of the number of genes in each cluster. Default is TRUE.
add2	add 2 clusters information.
cexCol	numbers, used as cex.axis in for the column axis labeling.
...	other parameters to heatmap.3.

Value

a gene expression heatmap with Cluster information figure

See Also

clEnrich, heatmap.3 and heatmapPEI

Examples

```
data(Psoriasis)
annofile <- system.file("extdata", "c2.cp.kegg.v7.01.symbols.gmt.xz",
package="cogena")

## Not run:
genecl_result <- coExp(DEexprs, nClust=2:3, clMethods=c("hierarchical","kmeans"),
    metric="correlation", method="complete", ncore=2, verbose=TRUE)

clen_res <- clEnrich(genecl_result, annofile=annofile, sampleLabel=sampleLabel)

#summay this cogena object
summary(clen_res)

#heatmapCluster

heatmapCluster(clen_res, "hierarchical", "3")
heatmapcol <- gplots::redgreen(75)
heatmapCluster(clen_res, "hierarchical", "3", heatmapcol=heatmapcol)

## End(Not run)
```

---

heatmapCmap *heatmap designed for for CMap gene set only*

Description

heatmapCmap is desgined for the cogena result from CMap only so as to collapse the multi-isntance drugs in CMap!

Usage

```
heatmapCmap(
  object,
  method = clusterMethods(object),
  nCluster = nClusters(object),
  orderMethod = "max",
  MultiInstance = "drug",
  CutoffNumGeneset = 20,
  CutoffPVal = 0.05,
  mergeMethod = "mean",
  low = "grey",
  high = "red",
  na.value = "white",
  maintitle = NULL,
  printGS = FALSE,
  add2 = TRUE,
  geom = "tile"
)

## S4 method for signature 'cogena'
heatmapCmap(
  object,
  method = clusterMethods(object),
  nCluster = nClusters(object),
  orderMethod = "max",
  MultiInstance = "drug",
  CutoffNumGeneset = 20,
  CutoffPVal = 0.05,
  mergeMethod = "mean",
  low = "grey",
  high = "red",
  na.value = "white",
  maintitle = "cogena",
  printGS = FALSE,
  add2 = TRUE,
  geom = "tile"
)
```

Arguments

object	a genecl or cogena object
method	as clMethods in genecl function
nCluster	as nClust in cogena function.
orderMethod	the order method, default is max, other options are "mean", "all", "I", "II" or a number meaning the ith cluster.
MultiInstance	merge multi instances. Options are "drug", "celldrug", "conccelldrug", "concdrug".

CutoffNumGeneset	the cut-off of the number of gene sets in the return table. The default is 20.
CutoffPVal	the cut-off of p-value. The default is 0.05.
mergeMethod	max or mean. The default is mean.
low	colour for low end of gradient.
high	colour for high end of gradient.
na.value	Colour to use for missing values.
maintitle	a character. Default is null
printGS	print the enriched gene set names or not. Default is FALSE
add2	enrichment score for add Up and Down reuglated genes.
geom	tile or circle

Details

orderMethod:

- max. ordered by the max value in clusters beside all
- mean. ordered by the mean value in clusters beside all
- All. ordered by all genes
- Up. ordered by up-regulated genes (add2 should be TRUE)
- Down. ordered by down-regulated genes (add2 should be TRUE)

MultiInstance:

- drug. merge based on cmap_name
- celldrug. merge based on cmap_name and cell type
- conccelldrug. merge based on cmap_name, cell type and concentration

Value

a gene set enrichment heatmap

Examples

```
data(Psoriasis)
annofile <- system.file("extdata", "CmapDn100.gmt.xz", package="cogena")
## Not run:
genecl_result <- coExp(DEexprs, nClust=3, clMethods=c("pam"),
    metric="correlation", method="complete", ncore=2, verbose=TRUE)
clen_res1 <- clEnrich(genecl_result, annofile=annofile, sampleLabel=sampleLabel)

heatmapCmap(clen_res1, "pam", "3", orderMethod="2")
heatmapCmap(clen_res1, "pam", "3", orderMethod="2", MultiInstance="concdrug")

## End(Not run)
```

heatmapPEI *heatmap of the gene set enrichment from a cogena object.*

Description

heatmap of the gene set enrichment score. After obtaining the ennrichemt of clusters for each gene set, the heatmapPEI will show it as a heatmap in order. The value shown in heatmapPEI is the -log2(fdr), representing the enrichment score.

Usage

```
heatmapPEI(
  object,
  method,
  nCluster,
  CutoffNumGeneset = 20,
  CutoffPVal = 0.05,
  orderMethod = "max",
  roundvalue = TRUE,
  low = "grey",
  high = "red",
  na.value = "white",
  maintitle = NULL,
  printGS = FALSE,
  add2 = TRUE,
  geom = "tile",
  wrap_with = 40
)

## S4 method for signature 'cogena'
heatmapPEI(
  object,
  method = clusterMethods(object),
  nCluster = nClusters(object),
  CutoffNumGeneset = 20,
  CutoffPVal = 0.05,
  orderMethod = "max",
  roundvalue = TRUE,
  low = "grey",
  high = "red",
  na.value = "white",
  maintitle = NULL,
  printGS = FALSE,
  add2 = TRUE,
  geom = "tile",
  wrap_with = 60
)
```

Arguments

object	a genecl or cogena object
method	as clMethods in genecl function
nCluster	as nClust in cogena function.
CutoffNumGeneset	the cut-off of the number of gene sets in the return table
CutoffPVal	the cut-off of p-value. The default is 0.05.
orderMethod	the order method, default is max, other options are "mean", "all", "I", "II" or a number meaning the ith cluster.
roundvalue	The default is TRUE. whether or not round the data. such as round(1.54, 1)=1.5
low	colour for low end of gradient.
high	colour for high end of gradient.
na.value	Colour to use for missing values.
maintitle	a character. like GSExxx. the output of figure will like "cogena: kmeans 3 GSExxx" in two lines. Default is NULL
printGS	print the enriched gene set names or not. Default is FALSE
add2	enrichment score for add Up and Down reuglated genes.
geom	tile or circle
wrap_with	default 40. wrap strings

Details

The x-axis shows cluster i and the number of genes in cluster, with red means cluster containing up-regulated genes, green means down-regulated genes, black means there are up and down regulated genes in this cluster and blue means all DEGs. If parameter add2 is true, another two columns will be shown as well, representing the up and down regulated genes.

The direction of DEGs are based on latter Vs former from sample labels. For example, labels are as.factor(c("ct", "Disease")), the "Disease" are latter compared with "ct". Usually, the order is the alphabet.

The y-axis represents the gene sets enriched.

orderMethod:

- max. ordered by the max value in clusters beside all
- mean. ordered by the mean value in clusters beside all
- All. ordered by all genes
- Up. ordered by up-regulated genes (add2 should be TRUE)
- Down. ordered by down-regulated genes (add2 should be TRUE)

Value

a gene set enrichment heatmap

See Also

clEnrich and heatmapCluster

Examples

```
data(Psoriasis)
annofile <- system.file("extdata", "c2.cp.kegg.v7.01.symbols.gmt.xz",
package="cogena")

## Not run:
genecl_result <- coExp(DEexprs, nClust=2:3, clMethods=c("hierarchical","kmeans"),
    metric="correlation", method="complete", ncore=2, verbose=TRUE)

clen_res <- clEnrich(genecl_result, annofile=annofile, sampleLabel=sampleLabel)

#summay this cogena object
summary(clen_res)

#heatmapPEI
heatmapPEI(clen_res, "kmeans", "2")
heatmapPEI(clen_res, "kmeans", "2", orderMethod="mean")
heatmapPEI(clen_res, "kmeans", "3", CutoffNumGeneset=20,
    low = "#132B43", high = "#56B1F7", na.value = "grey50")

## End(Not run)
```

---

mat *mat: get the original data from a genecl object.*

Description

mat: get the original data from a genecl object.

Usage

```
mat(object)

## S4 method for signature 'genecl'
mat(object)

## S4 method for signature 'cogena'
mat(object)
```

Arguments

object	a genecl or cogena object

Value

mat: a matrix

Examples

```
## Not run:
mat(genecl_result)

## End(Not run)
```

---

nClusters *nClusters: get the number of clusters from a genecl object.*

Description

nClusters: get the number of clusters from a genecl object.

Usage

```
nClusters(object)

## S4 method for signature 'genecl'
nClusters(object)

## S4 method for signature 'cogena'
nClusters(object)
```

Arguments

object	a genecl or cogena object

Value

nClusters: a numeric vector.

Examples

```
## Not run:
nClusters(genecl_result)

## End(Not run)
```

PEI *Significance of Gene sets enrichment.*

Description

Caculating the significance of Gene sets enrichment based on the hypergeometric test. This function is mainly used internally.

Usage

```
PEI(genenames, annotation, annotationGenesPop)
```

Arguments

genenames	a vector of gene names.
annotation	data.frame with the gene (like all the differentially expressed genes) in row, gene set in column.
annotationGenesPop	data.frame with the gene in row, gene set in column. Here genes are genes in population with filering the non-nformative genes better.

Details

Here the genes in annotation can be a varity of types. like all the DEG, up-regualted genes or genes in a cluster. the gene names should be consistent with the genes in the gene sets.

Value

a vector with P-values.

Examples

```
data(Psoriasis)
data(AllGeneSymbols)
annofile <- system.file("extdata", "c2.cp.kegg.v7.01.symbols.gmt.xz", package="cogena")
annoBG <- gene2set(annofile, AllGeneSymbols)
res <- PEI(rownames(DEexprs)[1:200], gene2set(annofile, rownames(DEexprs)[1:200]), annoBG)
```

Psoriasis *Psoriasis dataset.*

Description

An example dataset of Psoriasis. This dataset is used for illustration of the usage of cogena package. It has been normalised the expression profling using rma method, filtered some non-informative genes using MetaDE package and analysed the differentially expressed genes using limma package with the cut-off adjuested p-value 0.05 and abs(logFC) >=1.

Format

two objects: DEexprs and sampleLabel.

DEexprs expression of DEG. There are 706 DEGs and 116 samples.

sampleLabel the label of sample, There are 58 control and 58 Psoriasis.

Source

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE13355

sampleLabel *label of samples*

Description

label of samples

Format

a vector with 116 element.

Source

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE13355

show,cogena-method *show: show the class of cogena or genecl object*

Description

show: show the class of cogena or genecl object

Usage

```
## S4 method for signature 'cogena'
show(object)

## S4 method for signature 'genecl'
show(object)
```

Arguments

object	a genecl or cogena object

Value

show which instance

Examples

```
## Not run:
show(genecl_result)

## End(Not run)
```

sota *Self-organizing tree algorithm (SOTA)*

Description

Computes a Self-organizing Tree Algorithm (SOTA) clustering of a dataset returning a SOTA object. This function comes from sota in the clValid package with litter change.

Usage

```
sota(
  data,
  maxCycles,
  maxEpochs = 1000,
  distance = "euclidean",
  wcell = 0.01,
```

```
  pcell = 0.005,
  scell = 0.001,
  delta = 1e-04,
  neighb.level = 0,
  maxDiversity = 0.9,
  unrest.growth = TRUE,
  ...
)

## S3 method for class 'sota'
print(x, ...)

## S3 method for class 'sota'
plot(x, cl = 0, ...)
```

Arguments

data	data matrix or data frame. Cannot have a profile ID as the first column.
maxCycles	integer value representing the maximum number of iterations allowed. The resulting number of clusters returned by sota is maxCycles+1 unless unrest.growth is set to FALSE and the maxDiversity criteria is satisfied prior to reaching the maximum number of iterations
maxEpochs	integer value indicating the maximum number of training epochs allowed per cycle. By default, maxEpochs is set to 1000.
distance	character string used to represent the metric to be used for calculating dissimilarities between profiles. 'euclidean' is the default, with 'correlation' being another option.
wcell	alue specifying the winning cell migration weight. The default is 0.01.
pcell	value specifying the parent cell migration weight. The default is 0.005.
scell	value specifying the sister cell migration weight. The default is 0.001.
delta	value specifying the minimum epoch error improvement. This value is used as a threshold for signaling the start of a new cycle. It is set to 1e-04 by default.
neighb.level	integer value used to indicate which cells are candidates to accept new profiles. This number specifies the number of levels up the tree the algorithm moves in the search of candidate cells for the redistribution of profiles. The default is 0.
maxDiversity	value representing a maximum variability allowed within a cluster. 0.9 is the default value.
unrest.growth	logical flag: if TRUE then the algorithm will run maxCycles iterations regardless of whether the maxDiversity criteria is satisfied or not and maxCycles+1 clusters will be produced; if FALSE then the algorithm can potentially stop before reaching the maxCycles based on the current state of cluster diversities. A smaller than usual number of clusters will be obtained. The default value is TRUE.
...	Any other arguments.
x	an object of sota

cl	cl specifies which cluster is to be plotted by setting it to the cluster ID. By default, cl is equal to 0 and the function plots all clusters side by side.

Details

The Self-Organizing Tree Algorithm (SOTA) is an unsupervised neural network with a binary tree topology. It combines the advantages of both hierarchical clustering and Self-Organizing Maps (SOM). The algorithm picks a node with the largest Diversity and splits it into two nodes, called Cells. This process can be stopped at any level, assuring a fixed number of hard clusters. This behavior is achieved with setting the unrest.growth parameter to TRUE. Growth of the tree can be stopped based on other criteria, like the allowed maximum Diversity within the cluster and so on. Further details regarding the inner workings of the algorithm can be found in the paper listed in the Reference section.

Please note the 'euclidean' is the default distance metric different from sota

Value

A SOTA object.

data	data matrix used for clustering
c.tree	complete tree in a matrix format. Node ID, its Ancestor, and whether it's a terminal node (cell) are listed in the first three columns. Node profiles are shown in the remaining columns.
tree	incomplete tree in a matrix format listing only the terminal nodes (cells). Node ID, its Ancestor, and 1's for a cell indicator are listed in the first three columns. Node profiles are shown in the remaining columns.
clust	integer vector whose length is equal to the number of profiles in a data matrix indicating the cluster assingments for each profile in the original order.
totals	integer vector specifying the cluster sizes.
dist	character string indicating a distance function used in the clustering process.
diversity	vector specifying final cluster diverisities.

Author(s)

Vasyl Pihur, Guy Brock, Susmita Datta, Somnath Datta

References

Herrero, J., Valencia, A, and Dopazo, J. (2005). A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics, 17, 126-136.

Examples

```
#please ref the manual of sota function from clValid.
data(Psoriasis)

sotaCl <- sota(as.matrix(DEexprs), 4)
```

summary,genecl-method *summary: a summary of a genecl object.*

Description

summary: a summary of a genecl object.

Usage

```
## S4 method for signature 'genecl'
summary(object)

## S4 method for signature 'cogena'
summary(object)
```

Arguments

object	a genecl or cogena object

Value

summary: a summary of a genecl object.

Examples

```
## Not run:
summary(genecl_result)

## End(Not run)
```

upDownGene *Show up or down-regualted genes for a clustering method and the number of clusters.*

Description

The value means up or down regulated genes for each cluster. 1 suggests that genes in the cluster is up-regualted genes, while -1 down-regualted genes. value within (-1, 1) means genes there are both up and down regulated genes in the cluster. Return a vector with the length of nCluster if add2 is FALSE, or the length of nCluster + 2 if add2 is TRUE and nCluster is not 2. In the latter situation, the last two itemes represent Up and Down reuglated genes

Usage

```
upDownGene(object, method, nCluster, add2 = FALSE)

## S4 method for signature 'cogena'
upDownGene(object, method, nCluster, add2 = FALSE)

logfc(dat, sampleLabel)
```

Arguments

object	a genecl or cogena object
method	as clMethods in genecl function
nCluster	cluster number
add2	add2 enrichment score for add Up and Down reuglated genes
dat	gene expression data frame
sampleLabel	factor. sampleLabel with names

Value

upDownGene: a vector

logfc: a data.frame

Examples

```
data(Psoriasis)
annofile <- system.file("extdata", "c2.cp.kegg.v7.01.symbols.gmt.xz",
package="cogena")


genecl_result <- coExp(DEexprs, nClust=2:3, clMethods=c("hierarchical","kmeans"),
    metric="correlation", method="complete", ncore=2, verbose=TRUE)

clen_res <- clEnrich(genecl_result, annofile=annofile, sampleLabel=sampleLabel)

upDownGene(clen_res, "kmeans", "3", add2=TRUE)

upDownGene(clen_res, "kmeans", "2", add2=FALSE)
```

Index

∗ **cluster**
    sota, 31
∗ **datasets**
    AllGeneSymbols, 3
    DEexprs, 12
    Psoriasis, 30
    sampleLabel, 30
∗ **package**
    cogena_package, 10

AllGeneSymbols, 3
annotationListToMatrix *(*gene2set*)*, 14

clEnrich, 3, *8*, *12*, *17, 18*, *22*, *27*
clEnrich_one, 4
clusterMethods, 6
clusterMethods,cogena-method
        *(*clusterMethods*)*, 6
clusterMethods,cogena_methods
        *(*clusterMethods*)*, 6
clusterMethods,genecl-method
        *(*clusterMethods*)*, 6
clusterMethods,genecl_methods
        *(*clusterMethods*)*, 6
coExp, 7
cogean_package *(*cogena_package*)*, 10
cogena *(*cogena_package*)*, 10
cogena-class, 9
cogena_package, 10
corInCluster, 11
corInCluster,cogena-method
        *(*corInCluster*)*, 11
corInCluster,cogena_methods
        *(*corInCluster*)*, 11
corrplot, *11, 12*

DEexprs, 12

enrichment, 13
enrichment,cogena-method *(*enrichment*)*,
        13
enrichment,cogena_methods *(*enrichment*)*,
        13

gene2set, 14
genecl-class, 15
geneclusters, 16
geneclusters,cogena-method
        *(*geneclusters*)*, 16
geneclusters,cogena_methods
        *(*geneclusters*)*, 16
geneclusters,genecl-method
        *(*geneclusters*)*, 16
geneclusters,genecl_methods
        *(*geneclusters*)*, 16
geneExpInCluster, 17
geneExpInCluster,cogena-method
        *(*geneExpInCluster*)*, 17
geneExpInCluster,cogena_methods
        *(*geneExpInCluster*)*, 17
geneInCluster, 18
geneInCluster,cogena-method
        *(*geneInCluster*)*, 18
geneInCluster,cogena_methods,cluster_methods
        *(*geneInCluster*)*, 18
gmt2list, *15*, 19
gmtlist2file, 20

heatmap.3, *22*
heatmapCluster, 20, *27*
heatmapCluster,cogena-method
        *(*heatmapCluster*)*, 20
heatmapCmap, 22
heatmapCmap,cogena *(*heatmapCmap*)*, 22
heatmapCmap,cogena-method
        *(*heatmapCmap*)*, 22
heatmapPEI, *22*, 25
heatmapPEI,cogena *(*heatmapPEI*)*, 25
heatmapPEI,cogena-method *(*heatmapPEI*)*,
        25

logfc *(*upDownGene*)*, 34

logfc, *(*upDownGene*)*, 34

mat, 27
mat,cogena-method *(*mat*)*, 27
mat,cogena_methods *(*mat*)*, 27
mat,genecl-method *(*mat*)*, 27
mat,genecl_methods *(*mat*)*, 27

nClusters, 28
nClusters,cogena-method *(*nClusters*)*, 28
nClusters,cogena_methods *(*nClusters*)*, 28
nClusters,genecl-method *(*nClusters*)*, 28
nClusters,genecl_methods *(*nClusters*)*, 28

PEI, 29
plot.sota *(*sota*)*, 31
print.sota *(*sota*)*, 31
Psoriasis, 30
Psoriasis,DEexprs,sampleLabel,cogena_result
*(*Psoriasis*)*, 30

sampleLabel, 30
show,cogena-method, 31
show,cogena_methods
*(*show,cogena-method*)*, 31
show,genecl-method
*(*show,cogena-method*)*, 31
sota, 31, *31*, *33*
summary,cogena-method
*(*summary,genecl-method*)*, 34
summary,cogena_methods
*(*summary,genecl-method*)*, 34
summary,genecl-method, 34

upDownGene, 34
upDownGene,cogena *(*upDownGene*)*, 34
upDownGene,cogena-method *(*upDownGene*)*,
34