Package ‘cogena’

October 8, 2015

Version 1.0.0

Title co-expressed gene-set enrichment analysis

Description Description: Gene set enrichment analysis is a valuable tool for the study of molecular mechanisms that underpin complex biological traits. As the method is conventionally used on entire omic datasets, such as transcriptomes, it may be dominated by pathways and processes that are substantially represented in a dataset, however the approach may overlook smaller scale, but highly correlated cellular events that may be of great biological relevance. In order to detect these discrete molecular triggers, we developed a tool, co-expressed gene-set enrichment analysis (cogena), for clustering differentially expressed genes and identification of highly correlated molecular expression clusters. Cogena offers the user a range of clustering methods, including hierarchical clustering, model based clustering and self-organised mapping, based on different distance metrics like correlation and mutual information. After obtaining and visualising clusters, cogena performs gene set enrichment. These gene sets can be sourced from the Molecular Signatures Database (MSigDB) or user-defined gene sets. By performing gene set enrichment across expression clusters, we find considerable enhancement in the resolution of molecular signatures in omic data at the cluster level compared to the whole.

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

Depends R (>= 3.2), cluster, ggplot2, gplots, amap

Imports methods, class, foreach, parallel, doParallel, fastcluster, corrplot, reshape2, devtools

Suggests Biobase, annotate, kohonen, mclust, biwt, knitr

Collate dist_class.R cogena_class.R PD.R PEI.R cogena.R cogena_methods.R enrichment.R gene2set.R geneInCluster.R geneExpInCluster.R corInCluster.R gmt2list.R heatmap.3.R

heatmapCluster.R heatmapPEI.R heatmapPEI2.R hubgeneInCluster.R
optCluster.R sota.R vClusters.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

Author Zhilong Jia [aut, cre], Michael Barnes [aut]

Maintainer Zhilong Jia <zhilongjia@gmail.com>

R topics documented:

clusterMethods 3
cogena 4
cogena-class 6
corInCluster 7
DEexprs 8
enrichment 9
gene2set 10
geneExpInCluster 11
geneInCluster 12
gmt2list 13
heatmapCluster 13
heatmapPEI 15
heatmapPEI2 16
hubgeneInCluster 18
optCluster 19
PD 20
sampleLabel 20
sota 21

Index **24**

clusterMethods *Basic methods for a cogena object.*

Description

clusterMethods: get the methods of clustering used.

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

clusters: get the cluster of a certain clustering method.

mat: get the original data from a cogena object.

summary: a summary of a cogena object.

Usage

```
clusterMethods(object)

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

nClusters(object)

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

clusters(object, method)

## S4 method for signature 'cogena'
clusters(object, method = clusterMethods(object))

mat(object)

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

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

Arguments

`object`	a cogena object
`method`	as clMethods in cogena function

Value

clusterMethods: a character vector.

nClusters: a numeric vector.

clusters: a list or hclust depends on the method

mat: a matrix

summary: a summary of a cogena object.

Examples

```
data(PD)
annofile <- system.file("extdata", "c2.cp.kegg.v4.0.symbols.gmt",
package="cogena")
cogena_result <- cogena(DEexprs, nClust=2:3,
clMethods=c("hierarchical","kmeans"), metric="correlation",
method="complete",  annofile=annofile, sampleLabel=sampleLabel,
ncore=1, verbose=TRUE)
clusterMethods(cogena_result)
## Not run:
nClusters(cogena_result)

## End(Not run)
## Not run:
clusters(cogena_result, "kmeans")
clusters(cogena_result, "hierarchical")

## End(Not run)
## Not run:
mat(cogena_result)

## End(Not run)
## Not run:
summary(cogena_result)

## End(Not run)
```

---

cogena *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

```
cogena(obj, nClust, clMethods = "hierarchical", metric = "correlation",
  method = "complete", annofile = NULL, sampleLabel = NULL, ncore = 2,
  TermFreq = 0, 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", 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".

`annofile` gene set filename.

`sampleLabel` factor or character vector with names are sample names. only used for plotting.

`ncore` Number of core used. The default is 2.

`TermFreq` a value from [0,1) to filter low-frequence gene sets.

`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" 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 cogena object

Examples

```
data(PD)

#annotaion
annoGMT <- "c2.cp.kegg.v4.0.symbols.gmt"
annofile <- system.file("extdata", annoGMT, package="cogena")
#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"

# the cogena analysis
cogena_result <- cogena(DEexprs, nClust=nClust, clMethods=clMethods,
    metric=metric, method=method, annofile=annofile, sampleLabel=sampleLabel,
    ncore=ncore, verbose=TRUE)
```

`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.

`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

`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, nClusters, ith, corMethod = "pearson",
  plotMethod = "circle", type = "upper", ...)

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

Arguments

`object` a cogena object

`method` as clMethods in cogena function

`nClusters` as nClust in cogena function.

`ith` the ith cluster.

`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

`cogena` `corrplot`

Examples

```
data(PD)
annofile <- system.file("extdata", "c2.cp.kegg.v4.0.symbols.gmt",
package="cogena")
cogena_result <- cogena(DEexprs, nClust=c(2,10),
clMethods=c("hierarchical","kmeans"), metric="correlation",
method="complete",  annofile=annofile, sampleLabel=sampleLabel,
ncore=1, verbose=TRUE)
corInCluster(cogena_result, "kmeans", "10", "10")
corInCluster(cogena_result, "kmeans", "10", "10", plotMethod="square")
```

`DEexprs` *gene expression of DEG*

Description

gene expression of DEG

Format

matrix with 1243 DEGs (row) and 17 samples (column).

Source

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE20163

`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, nClusters, CutoffNumGeneset = Inf,
  CutoffPVal = 0.05, orderMethod = "max", roundvalue = TRUE)

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

Arguments

`object`	a cogena object
`method`	as clMethods in cogena function
`nClusters`	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

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)
- II. ordered by the II cluster in two clusters (Up or Down-regulated)
- a number. like 2, "3".

Value

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

Examples

```
data(PD)
data(PD)
annofile <- system.file("extdata", "c2.cp.kegg.v4.0.symbols.gmt",
package="cogena")
cogena_result <- cogena(DEexprs, nClust=2:3,
clMethods=c("hierarchical","kmeans"), metric="correlation",
method="complete",  annofile=annofile, sampleLabel=sampleLabel,
ncore=1, verbose=TRUE)
enrichment.table1 <- enrichment(cogena_result, "kmeans", "3")
enrichment.table2 <- enrichment(cogena_result, "kmeans", "3",
CutoffNumGeneset=10, orderMethod="mean")
```

---

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(PD)

#annotaion
annoGMT <- "c2.cp.kegg.v4.0.symbols.gmt"
annofile <- system.file("extdata", annoGMT, package="cogena")
# the DEG gene-sets matrix
anno <- gene2set(annofile, rownames(DEexprs))
```

`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, nClusters)

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

Arguments

`object`	a cogena object
`method`	as clMethods in cogena function
`nClusters`	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

`cogena`

Examples

```
data(PD)
annofile <- system.file("extdata", "c2.cp.kegg.v4.0.symbols.gmt",
package="cogena")
cogena_result <- cogena(DEexprs, nClust=2:3,
clMethods=c("hierarchical","kmeans"), metric="correlation",
method="complete",  annofile=annofile, sampleLabel=sampleLabel,
ncore=1, verbose=TRUE)
#summay this cogena object
summary(cogena_result)

#geneExpInCluster
geneExpInCluster(cogena_result, "kmeans", "3")
```

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, nClusters, ith)

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

Arguments

`object`	a cogena object
`method`	as clMethods in cogena function
`nClusters`	as nClust in cogena function.
`ith`	the ith cluster.

Value

a character vector containing the gene names.

See Also

cogena

Examples

```
data(PD)
annofile <- system.file("extdata", "c2.cp.kegg.v4.0.symbols.gmt",
package="cogena")
cogena_result <- cogena(DEexprs, nClust=2:3,
clMethods=c("hierarchical","kmeans"), metric="correlation",
method="complete",  annofile=annofile, sampleLabel=sampleLabel,
ncore=1, verbose=TRUE)
#summay this cogena object
summary(cogena_result)
```

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

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

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

Examples

```
anno <- "c2.cp.kegg.v4.0.symbols.gmt"
annofile <- system.file("extdata", anno, package="cogena")
gmt2list(annofile)
```

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

Description

heatmap of gene expression profilings with cluster-based color indication.

Usage

```
heatmapCluster(object, method, nClusters, sampleColor = c("darkblue", "cyan"),
  clusterColor = NULL, clusterColor2 = NULL, heatmapcol = NULL,
  maintitle = NULL, printSum = TRUE, ...)

## S4 method for signature 'cogena'
heatmapCluster(object, method = clusterMethods(object),
  nClusters = nClusters(object), sampleColor = c("darkblue", "cyan"),
  clusterColor = NULL, clusterColor2 = NULL, heatmapcol = NULL,
  maintitle = NULL, printSum = TRUE, ...)
```

Arguments

`object`	a cogena object
`method`	as clMethods in cogena function
`nClusters`	as nClust in cogena function.
`sampleColor`	a color vector with the sample length. The default is c("darkblue", "cyan").
`clusterColor`	a color vector with the cluster length. The default is rainbow(nClusters(object)).
`clusterColor2`	a color vector with 2 elements. The default is c("coral3", "deepskyblue1").
`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.
`...`	other parameters to heatmap.3.

Value

a gene expression heatmap with Cluster information figure

See Also

`cogena`, `heatmap.3` and `heatmapPEI`

Examples

```
data(PD)
annofile <- system.file("extdata", "c2.cp.kegg.v4.0.symbols.gmt",
package="cogena")
cogena_result <- cogena(DEexprs, nClust=2:3,
clMethods=c("hierarchical","kmeans"), metric="correlation",
method="complete",  annofile=annofile, sampleLabel=sampleLabel,
ncore=1, verbose=TRUE)

#summay this cogena object
summary(cogena_result)

#heatmapCluster
```

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

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

Description

heatmap of the gene set enrichment index. After obtaining the ennrichemt of clusters in the gene sets, the heatmapPEI will show it as a heatmap with order.

Usage

```
heatmapPEI(object, method, nClusters, CutoffNumGeneset = 20,
  CutoffPVal = 0.05, orderMethod = "max", roundvalue = TRUE,
  low = "green", high = "red", na.value = "white", maintitle = NULL,
  printGS = TRUE)

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

Arguments

`object` a cogena object

`method` as clMethods in cogena function

`nClusters` 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 TRUE.

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 only two clusters (Up or Down-regulated)
- II. ordered by the II cluster in only two clusters (Up or Down-regulated)

Value

a gene set enrichment heatmap

See Also

cogena and heatmapCluster

Examples

```
#' data(PD)
annofile <- system.file("extdata", "c2.cp.kegg.v4.0.symbols.gmt",
package="cogena")
cogena_result <- cogena(DEexprs, nClust=2:3,
clMethods=c("hierarchical","kmeans"), metric="correlation",
method="complete",  annofile=annofile, sampleLabel=sampleLabel,
ncore=1, verbose=TRUE)
#summay this cogena object
summary(cogena_result)

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

heatmapPEI2 *heatmap of the gene set enrichment_score matrix directly*

Description

heatmap of the gene set enrichment_score matrix directly. After obtaining the ennrichemt of clusters in the gene sets via enrichment, the heatmapPEI2 will show it as a heatmap.

Usage

```
heatmapPEI2(object, enrichment_score, method, nClusters, whichCluster,
  CutoffNumGeneset = 60, low = "grey", high = "red", na.value = "white",
  title = NULL)

## S4 method for signature 'cogena'
heatmapPEI2(object, enrichment_score, method, nClusters,
  whichCluster, CutoffNumGeneset = 60, low = "grey", high = "red",
  na.value = "white", title = NULL)
```

Arguments

object a cogena object

enrichment_score a returned value from enrichment function

method as clMethods in cogena function

nClusters as nClust in cogena function.

whichCluster which cluster should be based to filter. The format is "Cluster number # number of genes in clsuters", like "1#22". This can be obtained by heatmapCluster

CutoffNumGeneset the cut-off of the number of gene sets in the return table

low colour for low end of gradient.

high colour for high end of gradient.

na.value Colour to use for missing values.

title a character. like GSExxx. the output of figure will like "cogena: kmeans 3 GSExxx" in two lines. Default is NULL

Details

This function aims to heatmap the enrichment_score directly. This is helpful on condition that there are so many enriched gene sets and you can filter the enrichment_score based on a criteria, like just one cluster.

Value

a gene set enrichment heatmap

Examples

```
data(PD)
annofile <- system.file("extdata", "c2.cp.kegg.v4.0.symbols.gmt",
package="cogena")
cogena_result <- cogena(DEexprs, nClust=2:3,
clMethods=c("hierarchical","kmeans"), metric="correlation",
method="complete",  annofile=annofile, sampleLabel=sampleLabel,
ncore=1, verbose=TRUE)
summary(cogena_result)
```

```
enrichment.table <- enrichment(cogena_result, "kmeans", "3")

heatmapPEI2(cogena_result, enrichment.table, "kmeans", "3", "1")
```

hubgeneInCluster *Show hub gene names in certain cluster.*

Description

Show hub gene names in certain cluster.

Usage

```
hubgeneInCluster(object, method, nClusters, ith)

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

Arguments

object	a cogena object
method	as clMethods in cogena function
nClusters	as nClust in cogena function.
ith	the ith cluster.

Value

a character vector.

See Also

cogena and geneInCluster

Examples

```
data(PD)
annofile <- system.file("extdata", "c2.cp.kegg.v4.0.symbols.gmt",
package="cogena")
cogena_result <- cogena(DEexprs, nClust=2:3,
clMethods=c("hierarchical","kmeans"), metric="correlation",
method="complete",  annofile=annofile, sampleLabel=sampleLabel,
ncore=1, verbose=TRUE)
#summay this cogena object
summary(cogena_result)

#hubgeneInCluster
hubgeneInCluster(cogena_result, "kmeans", "3", "2")
```

optCluster *get the best clustering methods and the number of clusters*

Description

get the best clustering methods and the number of clusters, based that the number of gene sets which are signifigant should be maximum.

Usage

```
optCluster(object, based = "inTotal", ncores = object@ncore,
  CutoffPVal = 0.05)

## S4 method for signature 'cogena'
optCluster(object, based = "inTotal",
  ncores = object@ncore, CutoffPVal = 0.05)
```

Arguments

object	a cogena object
based	counting method. Default is "inTotal" to count all the clusters and I, II, All. Other options are "All", "I", "II".
ncores	cores used for caculating optCluster. Default is same as ncores used during cogena function, but it will be the same as number of cores machine has if ncores parameter is exceed it.
CutoffPVal	the cut-off of p-value. Default is 0.05.

Value

a score matrix

Examples

```
data(PD)
annofile <- system.file("extdata", "c2.cp.kegg.v4.0.symbols.gmt",
package="cogena")
cogena_result <- cogena(DEexprs, nClust=2:3,
clMethods=c("hierarchical","kmeans"), metric="correlation",
method="complete",  annofile=annofile, sampleLabel=sampleLabel,
ncore=1, verbose=TRUE)
summary(cogena_result)

score <- optCluster(cogena_result)
score <- optCluster(cogena_result, based="All")
```

PD *Parkinson's Disease dataset.*

Description

an example dataset of Parkinson's Disease. 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 p-value 0.05.

Format

three objects: DEexprs, sampleLabel and cogena_result.

DEexprs expression of DEG. There are 1243 DEGs and 17 samples.

sampleLabel the label of sample, There are 9 control and 8 PD.

cogena_result an example of cogena result.

Source

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE20163

sampleLabel *label of samples*

Description

label of samples

Format

a vector with 17 element.

Source

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE20163

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(PD)

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

Index

∗Topic **cluster**
  sota, 21
∗Topic **datasets**
  DEexprs, 8
  PD, 20
  sampleLabel, 20

annotationListToMatrix (gene2set), 10

clusterMethods, 3
clusterMethods,cogena-method (clusterMethods), 3
clusterMethods,cogena_methods (clusterMethods), 3
clusters (clusterMethods), 3
clusters,cogena-method (clusterMethods), 3
clusters,cogena_methods (clusterMethods), 3
cogena, 4, *8*, *11*, *12*, *14*, *16*, *18*
cogena-class, 6
corInCluster, 7
corInCluster,cogena-method (corInCluster), 7
corInCluster,cogena_methods (corInCluster), 7
corrplot, *8*

DEexprs, 8

enrichment, 9, *16*, *17*
enrichment,cogena-method (enrichment), 9
enrichment,cogena_methods (enrichment), 9

gene2set, 10
geneExpInCluster, 11
geneExpInCluster,cogena-method (geneExpInCluster), 11
geneExpInCluster,cogena_methods (geneExpInCluster), 11
geneInCluster, 12, *18*
geneInCluster,cogena-method (geneInCluster), 12
geneInCluster,cogena_methods (geneInCluster), 12
gmt2list, *10*, 13

heatmap.3, *14*
heatmapCluster, 13, *16*
heatmapCluster,cogena-method (heatmapCluster), 13
heatmapPEI, *14*, 15
heatmapPEI,cogena (heatmapPEI), 15
heatmapPEI,cogena-method (heatmapPEI), 15
heatmapPEI2, 16
heatmapPEI2,cogena (heatmapPEI2), 16
heatmapPEI2,cogena-method (heatmapPEI2), 16
hubgeneInCluster, 18
hubgeneInCluster,cogena-method (hubgeneInCluster), 18

mat (clusterMethods), 3
mat,cogena-method (clusterMethods), 3
mat,cogena_methods (clusterMethods), 3

nClusters (clusterMethods), 3
nClusters,cogena-method (clusterMethods), 3
nClusters,cogena_methods (clusterMethods), 3

optCluster, 19
optCluster,cogena (optCluster), 19
optCluster,cogena-method (optCluster), 19

PD, 20
PD,DEexprs,sampleLabel,cogena_result (PD), 20

plot.sota *(*sota*)*, 21
print.sota *(*sota*)*, 21

sampleLabel, 20
sota, *21*, 21, *22*
summary,cogena-method *(*clusterMethods*)*, 3