Package 'SPADEVizR'

April 28, 2016

Type Package

Title Visualization and analysis of SPADE clustering results.

Version 1.0-0

Author Guillaume GAUTREAU, Ludovic PLATON and Nicolas TCHITCHEK

Maintainer Nicolas TCHITCHEK <nicolas.tchitchek@gmail.com>, Guillaume GAUTREAU <guillaume.gautreau@free.fr> and Ludovic PLATON <platon.ludovic@gmail.com>

Description The Spanning tree Progression of Density normalized Events (SPADE) algorithm is powerful for mining flow and mass cytometry data. SPADE xxx. SPADEVizR is an R package dedicated to post-clustering analysis based on the . SPADEVizR is able to identify aboundant clusters (AC), differentially enriched clusters (DEC), and cluster correlating with phenotypical variables (CC). Moreover, SPADEVizR provides a large panel of visual representations such as parallel coordinates, phenotypical heatmap, stream graph representation, volcano plot representations. Through all these methods SPADEVizR constitutes a powerful pipeline for the analysis of cell clustering results provided by SPADE.

License GPL-3

Imports flowCore,
ggplot2,
igraph,
MASS,
gtools,
reshape2,
plyr

LazyData true

Suggests knitr

VignetteBuilder knitr

RoxygenNote 5.0.1

R topics documented:

abundantClustersViewer *abundantClustersViewer*

Description

Generate a barplot representation to compare cluster abondances

Usage

```
abundantClustersViewer(AC, cluster.size = TRUE, all.label = FALSE)
```

Arguments

AC	the AC object
cluster.size	a logicial specifing if points size are related to cell count or not (TRUE by default)
all.label	a logicial specifing if all cluster label must be show or just significant cluster

Details

By default, only abundant clusters are labeled, set parameter 'all.label = TRUE' to visualize all labels.

Value

a ggplot object

AC-class *Abundant Clusters (AC) class definition*

Description

AC is a S4 object containing the result of computeAC function.

Details

AC is a printable and a plotable object calling the abundantClustersViewer() fonction.

Slots

sample.names a character vector containing the samples names used

cluster.size a numeric vector containing number of cells (sum of all samples) for each cluster

use.percentages a logical specifying if computations was performed on percentage of cell abondance

method a character containing the name of the statistical test used to identify the AC

method.adjust a character containing the name of the multiple correction method used

th.pvalue a numeric vector with pvalue threshold

`th.mean` a numeric vector with mean threshold

`result` a data.frame with clusters in row. The first and second colunm contain the mean and the standard deviation and the third contain the pvalue. Finnaly the last columns is a logicial ("significance") describing if the 2 thresholds was reached or not

---

`boxplotViewer` *boxplotViewer*

Description

Generate a boxplot representation to compare the cell enrichment of biological conditions for each cluster

Usage

```
boxplotViewer(Results, conditions, clusters = NULL, use.percentages = TRUE,
  label = FALSE, violin = TRUE)
```

Arguments

`Results`	a SPADEResults or Results object
`conditions`	conditions a named vector providing the correspondence between a sample name (in rownames) and the condition of this sample : NA to exclude a sample from tests
`clusters`	a numerical vector containing the clusters to use in the representation
`use.percentages`	a logical specifying if the visualisation should be performed on percentage
`label`	a logical to show sample label or not (FALSE by default)
`violin`	xxx

Details

xxx

Value

a ggplot object

CC-class *Correlated Clusters (CC) class definition*

Description

CC is a S4 object containing the result of computeCC function.

Details

CC is a printable and a plotable object calling the correlatedClustersViewer() fonction

Slots

`cluster.size` a numeric vector containing number of cells (sum of all samples) for each cluster

`variable` a numeric vector containing the expression values of the variable

`use.percentages` a logical specifying if computations was performed on percentage of cell abondance

`method` a character containing the name of the statistical test used to identify the CC

`method.adjust` a character containing the name of the multiple correction method used

`th.pvalue` a numeric vector with pvalue threshold

`th.correlation` a numeric vector with correlation threshold (r)

`result` a three colmuns dataframe with clusters in row. The first colunm contains the coefficiant of correlation (r), the second contains the associated pvalue and the third a logical (significance) specifying if the two thresholds was reached or not.

classifyEnrichmentProfiles *classifyEnrichmentProfiles xxx*

Description

xxx

Usage

```
classifyEnrichmentProfiles(Results, method = "hierarchical_h",
  class.number = NULL, eigencell.correlation.th = 0.8,
  clique.correlation.th = 0.7, hierarchical.correlation.th = 0.8)
```

Arguments

`Results`	a Results or SPADEResults object
`method`	a character specifying the clustering method among one of those : 'hierarchical','kmeans','eigencell','clique'
`class.number`	a numeric specifying the number of classes needed when the method parameter choosen is either 'hierarchical_k' or 'kmeans'

`eigencell.correlation.th`
: a numeric (ignored if method is not 'eigencell') specifying the correlation threshold (in [0,1], 0.8 by default) in case of eigencell clustering

`clique.correlation.th`
: a numeric (ignored if method is not 'clique') specifying the correlation threshold (in [0,1], 0.7 by default) in case of clique clustering

`hierarchical.correlation.th`
: a numeric (ignored if method is not 'hierarchical_h') in [0,1]) specifying the threshold of correlation (in [0,1], 0.8 by default) use to cut the hirerchical tree

Details

xxx

Value

a EnrichmentProfiles object

`classifyPhenoProfiles` *classifyPhenoProfiles xxx*

Description

xxx

Usage

```
classifyPhenoProfiles(Results, method = "hierarchical_h",
  class.number = NULL, eigencell.correlation.th = 0.8,
  clique.correlation.th = 0.7, hierarchical.correlation.th = 0.8)
```

Arguments

`Results`
: a Results or SPADEResults object

`method`
: a character specifying the clustering method among one of those : 'hierarchical','kmeans','eigencell','clique'

`class.number`
: a numeric specifying the number of classes needed when the method parameter choosen is either 'hierarchical_k' or 'kmeans'

`eigencell.correlation.th`
: a numeric (ignored if method is not 'eigencell') specifying the correlation threshold (in [0,1], 0.8 by default) in case of eigencell clustering

`clique.correlation.th`
: a numeric (ignored if method is not 'clique') specifying the correlation threshold (in [0,1], 0.7 by default) in case of clique clustering

`hierarchical.correlation.th`
: a numeric (ignored if method is not 'hierarchical_h') in [0,1]) specifying the threshold of correlation (in [0,1], 0.8 by default) use to cut the hirerchical tree

Details

xxx

Value

a PhenoProfiles object

`clusterViewer` *ClusterViewer*

Description

Cluster viewer

Usage

```
clusterViewer(Results, samples = NULL, clusters = NULL, markers = NULL,
  show.mean = "both")
```

Arguments

`Results`	a SPADEResuts or Result object (without quantiles bounds if a Results object is provided)
`samples`	a named vector providing the correspondence between samples name (in rownname) and the logical value TRUE to use these samples (all samples by default)
`clusters`	a character vector containing the clusters to use for the representation
`markers`	a pattern describing markers to observe
`show.mean`	a character : "none" "both" "only", "both" by default

Details

xxx

Value

a list of ggplot objects

`computeAC` *Compute the Abundant Clusters*

Description

Abundant Clusters are clusters wich have an enrichement significatyvely different of zeor

Usage

```
computeAC(Results, condition, use.percentages = TRUE, method = "t.test",
  method.adjust = NULL, th.pvalue = 0.05, th.mean = 0)
```

Arguments

Results	a Results or SPADEResults object
condition	a named vector providing the correspondence between a sample name (in rowname) and the logical value TRUE to test abondance for this sample or FALSE otherwise
use.percentages	a logical specifying if the computations should be performed on percentage
method	a character containing the statistical method to use for the ACs detection. The parameter can take the values "t.test" or "wilcox.test"
method.adjust	a character specifying if the pvalues should be corrected using the argument "method" available for function p.adjust, among : "holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr"
th.pvalue	a numeric specifying the pvalue threshold (0.05 by default)
th.mean	a numeric specifying the mean threshold (0 by default)

Details

xxx

Value

a AC object

---

computeCC *Compute the correlation of SPADE cluster with a cynetics phenotype*

Description

Correlated Clusters are clusters with count or

Usage

```
computeCC(Results, variable, use.percentages = TRUE, method = "pearson",
  method.adjust = NULL, th.pvalue = 0.05, th.correlation = 0.5)
```

Arguments

Results	a Results or SPADEResults object
variable	a numerical named vector providing the correspondence between a sample name (in rowname) and the specific phenotype or NA to ignore a sample
use.percentages	a logical specifying if the computations should be performed on percentage
method	a character indicating the correlation method to use : "pearson", "spearman"
method.adjust	a character specifying if the pvalues should be corrected using the argument "method" available for function p.adjust, among : "holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr"
th.pvalue	a numeric specifying the pvalue threshold (0.05 by default)
th.correlation	a numeric specifying the r two sided threshold (0.5 by default) in [0,1]

Details

xxx

Value

a CC object

`computeClique` *Internal - computeClique*

Description

xxx

Usage

```
computeClique(data, clique.correlation.th = 0.7)
```

Arguments

`data`	a matrix with all clusters in rownames
`eigencell.correlation.th`	

Details

xxx

Value

a dataframe containing the cluster ID, the classe of this cluster and a clustering score.

`computeDEC` *Compute the Differentially Enriched Clusters*

Description

Differentially Enriched Clusters are clusters for which the means of cell number are significantly different between two conditions.

Usage

```
computeDEC(Results, conditions, use.percentages = TRUE, method = "t.test",
  method.adjust = NULL, method.paired = FALSE, th.pvalue = 0.05,
  th.fc = 1)
```

Arguments

conditions	a named vector providing the correspondence between a sample name (in rownames) and the condition of this sample : NA to exclude a sample from tests, 1 or 2 to attribute this sample, respectively to the first or second condition
use.percentages	a logical specifying if the computations should be performed on percentage
method	a character specifying the name of the statistical test to use "t.test" or "wilcox.test"
method.adjust	a character specifying if the pvalues should be corrected using the argument "method" available for function p.adjust, among : "holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr"
method.paired	a logical indicating whether data measurement are paired (by default FALSE)
th.pvalue	a numeric specifying the pvalue threshold (0.05 by default)
th.fc	a numeric specifying the foldchange threshold (1 by default)
result	a Results or SPADEResults object

Details

xxx

Value

a DEC object

---

computeEigenCellClusters

Internal - computeEigenCellClusters

Description

xxx

Usage

```
computeEigenCellClusters(data, eigencell.correlation.th = 0.8)
```

Arguments

data	a matrix with all clusters in rownames
eigencell.correlation.th	

Details

xxx

Value

a dataframe containing the cluster ID, the classe of this cluster and a clustering score.

computeHierarchicalClustering *Internal - computeHierarchicalClustering*

Description

xxx

Usage

```
computeHierarchicalClustering(data, class.number = NULL,
  hierarchical.correlation.th = 0.8)
```

Arguments

data	a matrix with all clusters in rownames
class.number	the number of classe to cluster, if class.number is NULL numer of classe will be determined base on hierarchical correlation threshold
hierarchical.correlation.th	

Details

xxx

Value

a dataframe containing the cluster ID, the classe of this cluster and a clustering score.

computeKMeans *Internal - computeKMeans*

Description

xxx

Usage

```
computeKMeans(data, k)
```

Arguments

data	a matrix with all clusters in rownames
k	number of classes

Details

xxx

Value

a dataframe containing the cluster ID, the classe of this cluster and a clustering score.

computePhenoTable *Internal - computePhenoTable*

Description

computePhenoTable

Usage

```
computePhenoTable(SPADEResults, num = 5)
```

Arguments

SPADEResults	a SPADEResults object
num	a numeric indicating the number of "phenotype"

Details

xxx

Value

xxx

computeQuantile *Internal - Compute quantile with FCS flowset marker by marker*

Description

This function is used internally

Usage

```
computeQuantile(flowset, probs = c(0.05, 0.95))
```

Arguments

flowset	a flowCore flowset
probs	a vector of probabilities with 2 values in [0,1] to compute quantiles

Details

xxx

Value

a numeric matrix with bounds

computeQuantile.heuristic

Internal - Compute quantile with FCS flowset sample by sample

Description

This function is used internally, it provide the mean of quantiles from each sample to seed up computation

Usage

```
computeQuantile.heuristic(flowset, probs = c(0.05, 0.95))
```

Arguments

flowset	a flowCore flowset
probs	a vector of probabilities with 2 values in [0,1] to compute quantiles

Value

a numeric matrix with bounds

correlatedClustersViewer

correlatedClustersViewer

Description

Generate a representation to compare cluster abondances

Usage

```
correlatedClustersViewer(CC, cluster.size = TRUE, all.label = FALSE)
```

Arguments

CC	the CC object
cluster.size	a logicial specifing if points size are related to cell count or not (TRUE by default)
all.label	a logicial specifing if all cluster label must be show or just significant cluster

Details

xxx

Value

a ggplot object

DEC-class *Differentially Enriched Clusters (DEC) class definition*

Description

DEC is a S4 object containing the result of computeDEC function.

Details

DEC is a printable and a plotable object calling the volcanoViewer() fonction.

Slots

`sample.cond1` a character specifying the names of the samples corresponding the first condition

`sample.cond2` a character specifying the names of the samples corresponding the second condition

`cluster.size` a numeric vector containing number of cells (sum of all samples) for each cluster

`use.percentages` a logical specifying if computations was performed on percentage of cell abondance

`method` a character containing the name of the statistical test used to identify the DEC

`method.adjust` a character containing the name of the multiple correction method used

`method.paired` a logical indicating if the test has beeen performed in a paired manner

`th.pvalue` a numeric vector with pvalue threshold

`th.fc` a numeric vector with foldchange threshold

`result` a dataframe with clusters in row. The first and second colunm contain the mean and the standard deviation for the first condition, the third and fourth the mean and standard deviation for the second condition, the sixth the fold-change and the seventh the pvalue and finnaly the significance according to pvalue threshold and fold change threshold

dendro *title Internal - Build a dendrograms plot*

Description

xxx

Usage

```
dendro(ddata, row = !col, col = !row)
```

Arguments

`ddata`	xxx
`row`	xxx
`col`	xxx

Details

xxx

Value

a dendroplot

`distogramViewer` *biplotViewer*

Description

Generates a biplot representation with two markers

Generates a biplot representation with two markers

Generates a biplot representation with two markers

Generate a distogram representation showing the marker co-expression.

Usage

```
distogramViewer(Results)
```

Arguments

`Results`	a SPADEResults or Results object
`SPADEResults`	a SPADEResults object (Results object is not accepted)
`x.marker1`	a character indicating the marker name of the first dimension
`y.marker2`	a character indicating the marker name of the second dimension
`samples`	xxx
`clusters`	xxx
`default.min`	a numeric value indicating the lower bound of the biplot representation
`SPADEResults`	a SPADEResults object (Results object is not accepted)
`x.marker1`	a character indicating the marker name of the first dimension
`y.marker2`	a character indicating the marker name of the second dimension
`samples`	xxx
`clusters`	xxx
`default.min`	a numeric value indicating the lower bound of the biplot representation
`return.gg`	a logical indicating if the function should return a list of ggplot objects
`SPADEResults`	a SPADEResults object (Results object is not accepted)
`x.marker1`	a character indicating the marker name of the first dimension
`y.marker2`	a character indicating the marker name of the second dimension
`samples`	xxx
`clusters`	xxx
`default.min`	a numeric value indicating the lower bound of the biplot representation
`sample.merge`	xxx
`resample.ratio`	xxx

Details

In such representation, each dot corresponds to a cell profile and dot are ploted in a 2-dimentional space corresponding to the marker expressions.

In such representation, each dot corresponds to a cell profile and dot are ploted in a 2-dimentional space corresponding to the marker expressions.

In such representation, each dot corresponds to a cell profile and dot are ploted in a 2-dimentional space corresponding to the marker expressions.

xxx

Value

if return.gg is TRUE, the function returns a list of ggplot objects

if return.gg is TRUE, the function returns a list of ggplot objects

if return.gg is TRUE, the function returns a list of ggplot objects

a list of ggplot objects

`EnrichmentProfiles-class`

EnrichmentProfiles class definition

Description

EnrichmentProfiles is a S4 object containing the result of classifyEnrichmentProfiles() function. It classify clusters by their cells enrichment properties across the samples.

Details

EnrichmentProfiles is a printable and a plotable object calling the profileViewer() fonction

Slots

`method` a character specifying the method used to classify cluster

`method.parameter` a list of parameters used by the selected method

`cluster.size` a numeric vector with the number of cell in each cluster (sum of all samples)

`cluster.number` a numeric providing the number of cluster

`class.number` a numeric providing the number of classes

`classes` a two column dataframe with the cluster in first colunm and corresponding classe in the second colunm

exclude.markers *Internal - Removing of cell markers to exclude from a matrix*

Description

This function is used internally to remove one or several cell markers.

Usage

```
exclude.markers(data, exclude, colnames.FCS = NULL)
```

Arguments

data	a numeric matrix or flowset
exclude	a character vector containing the cell markers to be excluded
colnames.FCS	a character vector containing colnames if data is a FCS flowset

Details

xxx

Value

a numeric matrix without the cell markers to exclude

export *Exportation of SPADEVizR objects*

Description

Exports an object into a TAB delimeted file with parameters in header (begining with a sharp '#')

Usage

```
export(object, filename)

## S4 method for signature 'Results'
export(object, filename = "Results.txt")

## S4 method for signature 'AC'
export(object, filename = "AC.txt")

## S4 method for signature 'DEC'
export(object, filename = "DEC.txt")

## S4 method for signature 'CC'
export(object, filename = "CC.txt")

## S4 method for signature 'PhenoProfiles'
```

```
export(object, filename = "PhenoProfiles.txt")

## S4 method for signature 'EnrichmentProfiles'
export(object,
  filename = "EnrichmentProfiles.txt")
```

Arguments

object	a SPADEVizR object
filename	a character indicating the location of output file

Value

none

`filter.medians` *Internal - filter medians to exclude from a matrix*

Description

This function is used internally to remove raw or transform medians from SPADE matrix. CVS medians are always remove

Usage

```
filter.medians(data, use.raw.medians = FALSE)
```

Arguments

data	a SPADE matrix
use.raw.medians	a logicial specifying if "transformed" or "raw" medians will be use (FALSE by default)

Details

xxx

Value

a numeric matrix without the cell markers to exclude

generateReport *Generate report and plot.*

Description

Generate a report based on the SPADE result object.

Usage

```
generateReport(Results, PDFfile, reports = c("pheno", "kinetic", "cluster",
  "kinetic_cluster", "tree", "disto", "stream", "MDS"), clusters = NULL,
  markers = NULL, assignments = NULL, stat.objects = list(),
  profile.objects = list(), width = 29.7, height = 21)
```

Arguments

Results	a SPADEResuts or Result object
PDFfile	a character specifying the output path
reports	a vector of the plot names to add in the report (following the vector order). Plot names are : "pheno", "MDS", "cluster", "tree", "disto","stream". By default all plots will be add
clusters	a character vector of ID cluster to be reported
markers	a character vector of markers to be reported
stat.objects	a list of stat.object (object of type DEC, AC or CC)
profile.objects	a list of profile.objects (object of type PhenoProfiles or EnrichmentProfiles)
width	a numeric specifying the plot width
height	a numeric specifying the plot height

ggheatmap *title Internal - Create a list of elements allowing to build a heatmap*

Description

Use the function ggheatmap.plot to display this heatmap

Usage

```
ggheatmap(mat, dendrogram.type = "rectangle", num = 5,
  clustering.markers = NULL)
```

Arguments

mat	a matrix
dendrogram.type	a caracter spycifing the look of dendrograms ("rectangle" or "triangle", "rectangle" by default)
num	xxx
clustering.markers	a character vector of clustering markers

Value

a list of 3 plots (top dendrogram, right dendrogram, heatmap)

`ggheatmap.plot` *title Internal - ggheatmap.plot*

Description

ggheatmap.plot display the heatmap build by ggheatmap

Usage

```
ggheatmap.plot(L, col.width = 0.15, row.width = 0.15)
```

Arguments

`L`	the list of ggplot object provided by ggheatmap
`col.width`	size of horizontal dendrogram
`row.width`	size of vertical dendrogram

Details

xxx

Value

a ggplot2 axis

`g_axis` *title Internal - g_axis*

Description

Extract axis from a plot

Usage

```
g_axis(a.gplot, x = !y, y = !x)
```

Arguments

`a.gplot`	a ggplot2 plot
`x`	a logical, TRUE to extract x axis
`y`	a logical, TRUE to extract y axis

Details

x and y are mutuality excluded (both can't be TRUE) with priority to x

Value

a ggplot2 axis

g_legend *title Internal - g_legend*

Description

Extract legend from a plot

Usage

```
g_legend(a.gplot)
```

Arguments

a.gplot	a ggplot2 plot

Details

xxx

Value

a ggplot2 legend

importResults *importSPADEResults*

Description

Import the SPADE results from a specified path to a SPADEResult object.

Generate a SPADEresults object based on SPADE results.

The tables are loaded, the SPADE tree is loaded as well as the layout

Usage

```
importResults(marker.expressions, cells.count)
```

Arguments

marker.expressions	a numerical dataframe containing median expression values for each marker of each sample, in additions of markers, the 2 two first columns are "cluster" and "sample"
cells.count	a matrix of cells abondances with clusters in row and samples in column

Details

We advice to use quantile.heuristic = TRUE because it is faster and much more efficient in term of memory with a low impact on precision.

Value

Results a result object

`importSPADEResults` *importSPADEResults*

Description

Import the SPADE results from a specified path to a SPADEResult object.

Generate a SPADEresults object based on SPADE results.

The tables are loaded, the SPADE tree is loaded as well as the layout

Usage

```
importSPADEResults(path, dict = data.frame(), exclude.markers = NULL,
  probs = c(0.05, 0.95), use.raw.medians = FALSE,
  quantile.heuristic = FALSE)
```

Arguments

`path`	the a character specify the path of SPADE results folder
`dict`	a two column data.frame providing the correspondance between the original marker names (first column) and the new marker names (second column)
`exclude.markers`	a list of markers to exclude
`probs`	a vector of probabilities with 2 values in [0,1] to compute quantiles. First is the lower bound and second is the lower bound.
`use.raw.medians`	a logicial specifying if "transformed" or "raw" medians will be use (FALSE by default)
`quantile.heuristic`	a logicial specifying if quantile are compute for with all cells (FALSE), or is the means of the quantile of each samples (TRUE)

Details

We advice to use quantile.heuristic = TRUE because it is faster and much more efficient in term of memory with a low impact on precision.

Value

SPADEResults a SPADE result object

kineticsViewer *Kinetic Viewer*

Description

Kinetic viewer aim to represent the kinetics of xxx

Usage

```
kineticsViewer(Results, assignments, clusters = NULL,
  use.percentages = TRUE)
```

Arguments

Results	a SPADEResults or Results object
assignments	a 2 column data.frame with the samples names in rownames providing firstly the timepoints (numeric) and secondly the individuals (caracter) of the experiment
clusters	a numerical vector containing the clusters to use in the representation, by default all clusters will be use
use.percentages	a logical specifying if the visualisation should be performed on percentage

Details

xxx

Value

a ggplot object

MDSViewer *MDS viewer.*

Description

Generate a MDS representation based on the SPADE result object.

Usage

```
MDSViewer(Results, use.percentages = TRUE, assignments, clusters = NULL,
  space = "clusters", dist.method = "euclidean")
```

Arguments

Results	a SPADEResults or Results object
use.percentages	a logical specifying if the visualisation should be performed on percentage
assignments	a 2 column data.frame with the samples names in rownames providing firstly the timepoints (numeric) and secondly the individuals (caracter) of the experiment
clusters	specify the set of clusters to use
space	a caracter specifying the space ("clusters" or "samples", cluster by default)
dist.method	a character string containing the name of the distance measure to use

Details

xxx

Value

a list of ggplot objects

`names` *Results object's name*

Description

Named a Results object.

Named a SPADEResults object.

Named a AC object.

Named a DEC object.

Named a CC object.

Named a PhenoProfiles object.

Named a EnrichmentProfiles object.

Usage

```
## S4 method for signature 'Results'
names(x)

## S4 method for signature 'SPADEResults'
names(x)

## S4 method for signature 'AC'
names(x)

## S4 method for signature 'DEC'
names(x)

## S4 method for signature 'CC'
names(x)

## S4 method for signature 'PhenoProfiles'
names(x)

## S4 method for signature 'EnrichmentProfiles'
names(x)
```

Arguments

x	a Results object
x	a SPADEResults object
x	a AC object

x	a DEC object
x	a CC object
x	a PhenoProfiles object
x	a EnrichmentProfiles object

Value

none

none

none

none

none

none

none

`PhenoProfiles-class` *PhenoProfiles class definition*

Description

PhenoProfiles is a S4 object containing the result of classifyPhenoProfiles() function. It classify clusters by their markers properties provided by the fonction computePhenoTable()

Details

PhenoProfiles is a printable and a plotable object calling the xxx() fonction

Slots

`method` a character specifying the method used to classify cluster

`method.parameter` a list of parameters used by the selected method

`cluster.size` a numeric vector with the number of cell in each cluster (sum of all samples)

`cluster.number` a numeric providing the number of cluster

`class.number` a numeric providing the number of classes

`classes` a two column dataframe with the cluster in first colunm and corresponding classe in the second colunm

phenoViewer *Pheno Viewer*

Description

Heatmap Viewer aims to xxx

Usage

```
phenoViewer(SPADEResults, pheno.table = NULL, num = 5)
```

Arguments

SPADEResults	a SPADEResults object (Results object is not accepted)
pheno.table	a result of the function computePhenoTable
num	a numeric indicating the precision of computed expression scores

Details

xxx

Value

a list of ggplot objects

plot *xxx*

Description

xxx

Usage

```
plot(x, ...)

## S4 method for signature 'DEC'
plot(x, ...)

## S4 method for signature 'AC'
plot(x, ...)

## S4 method for signature 'CC'
plot(x, ...)

## S4 method for signature 'PhenoProfiles'
plot(x, ...)

## S4 method for signature 'EnrichmentProfiles'
plot(x, ...)
```

Arguments

xxx

Details

xxx

Value

xxx

---

print *Textual preview for Results objects*

Description

Prints a preview for a Results object.

Prints a preview for a SPADEResults object.

Prints a preview for a Abundant Clusters (AC) object.

Prints a preview for a Differentially Enriched Clusters (DEC) object.

Prints a preview for a Correlated Clusters (CC) object.

Prints a preview for a PhenoProfiles object.

Prints a preview for a EnrichmentProfiles object.

Usage

```
## S4 method for signature 'Results'
print(x)

## S4 method for signature 'SPADEResults'
print(x)

## S4 method for signature 'AC'
print(x)

## S4 method for signature 'DEC'
print(x)

## S4 method for signature 'CC'
print(x)

## S4 method for signature 'PhenoProfiles'
print(x)

## S4 method for signature 'EnrichmentProfiles'
print(x)
```

Arguments

x	a Results object
x	a SPADEResults object
x	a Abundant Clusters (AC) object
x	a Differentially Enriched Clusters (DEC) object
x	a Correlated Clusters (CC) object
x	a PhenoProfiles object
x	a EnrichmentProfiles object

Value

none

none

none

none

none

none

none

`profilesViewer` *profilesViewer*

Description

Generate a graph representation of PhenoProfiles classes

Generate a graph representation of PhenoProfiles classes

Usage

```
profilesViewer(profile.object, cluster.size = TRUE)
```

Arguments

`profile.object`	a PhenoProfiles object or an EnrichmentProfiles object
`cluster.size`	a logicial specifing if points size are related to cell count or not (TRUE by default)
`show.unclassified`	a logical specifying if unclassified clusters must be shown or not (FALSE by default)
`profile.object`	a PhenoProfiles object or an EnrichmentProfiles object
`cluster.size`	a logicial specifing if points size are related to cell count or not (TRUE by default)

Details

xxx

xxx

Value

a ggplot object

a ggplot object

`rename.markers` *Internal - Renaming cell markers*

Description

This function is used internally to rename the cell markers based on a dictionary.

Usage

```
rename.markers(header, dictionary)
```

Arguments

`header`	a character vector containing the original maker names
`dictionary`	a character vector containing a correspondence between the original and the new marker names

Details

dictionary is a data.frame used to rename the marker names. The first column must correspond to the original marker names, the second column must correspond to the new marker names.

Value

a character vector containing the renamed marker names

`Results-class` *Results class definition*

Description

Results is a S4 object containing cytometric results.

Details

Results is the core (super classe) of SPADEResults object but allow importation of data comming from other software than SPADE. It aims to store the data comming from the importResults() function.

Slots

`marker.expressions` a dataframe containing the number of cells for each cluster of each sample

`cells.count` a numerical dataframe containing marker median expressions for each cluster of each sample

`sample.names` a character vector containing the sample names

`marker.names` a character vector containing the markers names

`cluster.number` a numeric specyfing the number of clusters

show *Textual preview for Results objects*

Description

Show a preview a Results object.

Show a preview a SPADEResults object.

Show a preview a for Abundant Clusters (AC) object

Show a preview a for Differentially Enriched Clusters (DEC) object

Show a preview a Correlated Clusters (CC) object.

Show a preview a PhenoProfiles object.

Show a preview a EnrichmentProfiles object.

Usage

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

## S4 method for signature 'SPADEResults'
show(object)

## S4 method for signature 'AC'
show(object)

## S4 method for signature 'DEC'
show(object)

## S4 method for signature 'CC'
show(object)

## S4 method for signature 'PhenoProfiles'
show(object)

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

Arguments

x	a Results object
x	a SPADEResults object
x	a Abundant Clusters (AC) object
x	a Differentially Enriched Clusters (DEC) object
x	a Correlated Clusters (CC) object
x	a PhenoProfiles object
x	a EnrichmentProfiles object

Value

none

none

none

none

none

none

none

SPADEResults-class *SPADEResults class definition*

Description

SPADEResults is a S4 object containing SPADE results.

Details

SPADEResults extend Results object adding specific SPADE data. SPADEResults object is the central element used by the most of SPADEViR functions, it aims to store the SPADE data comming from the importSPADEResults() function.

Slots

`use.raw.medians` a logical specifying if the marker expressions are obtained from "raw_median" or "transformed" columns

`dictionary` a two column data.frame providing the correspondence between the original marker names (first column) and the new marker names (second column)

`marker.clustering` a logical vector specifying marker that have been used during the clustering precedure

`fcs.files` a character vector containing the absolute path of the original FCS files

`quantiles` a numerical matrix containing the quantiles for each each markers of cluster

`graph` a igraph object containing the SPADE tree

`graph.layout` a numerical matrix containing the layout of the SPADE tree

statSteamgraph *title Internal - Transforms data for a steam graph (from by ggTimeSeries : https://github.com/Ather-Energy/ggTimeSeries)*

Description

xxx

Usage

```
statSteamgraph
```

Arguments

xxx

Format

An object of class `statSteamgraph` (inherits from `Stat`, `ggproto`) of length 4.

Details

xxx

Value

xxx

stat_steamgraph *title Internal - Plot a steamgraph (from by ggTimeSeries : https://github.com/Ather-Energy/ggTimeSeries)*

Description

xxx

Usage

```
stat_steamgraph(mapping = NULL, data = NULL, show.legend = NA,
  inherit.aes = TRUE, na.rm = TRUE, ...)
```

Arguments

xxx

Details

xxx

Value

xxx

streamgraphViewer *streamgraphViewer*

Description

Generate a streamgraph representation showing the evolution of cell count in clusters across samples

Usage

```
streamgraphViewer(Results, order = NULL, clusters = NULL)
```

Arguments

Results	a SPADEResults or Results object
order	a named vector a named vector providing the correspondence between a sample name (in rownames) and an integer ordering samples in numeric (NA to exclude this sample)
clusters	a numerical vector containing the clusters to use in the representation, by default all clusters will be use

Details

xxx

Value

a ggplot object

treeViewer *SPADE Tree viewer*

Description

xxx

Usage

```
treeViewer(SPADEResults, samples = NULL, stat.object = NULL,
  vertex_size = c(1, 15))
```

Arguments

SPADEResults	a SPADEResults object (Results object is not accepted)
samples	a named vector providing the correspondence between samples name (in rowname) and the logical value TRUE to use these samples for cell counting (all samples by default)
stat.object	an AC, DEC or CC object to highligth significant clusters in the SPADE tree
vertex_size	a numeric vector of two values indicating the range of

Details

xxx

Value

a list of ggplot objects

`volcanoViewer` *Volcano Plot Viewer*

Description

Generate a Volcano plot representation based on Differentially Enriched Clusters (DEC) of The SPADE result object.

Usage

```
volcanoViewer(DEC = NULL, fc.log2 = TRUE, cluster.size = TRUE,
  all.label = FALSE)
```

Arguments

`DEC`	object DEC provided by the function computeDEC
`fc.log2`	a logicial specifing if foldchange or log2(foldchange) is use
`cluster.size`	a logicial specifing if points size are related to cell count or not (TRUE by default)
`all.label`	a logicial specifing if all cluster labels must be show or just significant cluster
`SPADEResults`	the SPADEViewer result object

Details

xxx

Value

a ggplot object

Index