Package 'SPADEVizR'

May 2, 2016

Type Package

Title Visualization and statistical analyses of cell clustering results generated by the SPADE clustering algorithm.

Version 1.0-0

Author Guillaume GAUTREAU and Nicolas TCHITCHEK

Maintainer Nicolas TCHITCHEK <nicolas.tchitchek@gmail.com> and Guillaume GAUTREAU <guillaume.gautreau@free.fr>

Description Flow and mass cytometry are experimental techniques used for the characterization of cell phenotypes. With the increase of usable cell markers (up to 50 markers), the identification of cell populations through manual gating is impossible. These high-dimensional data require new computational algorithms to automatically identify cell clusters. The SPADE algorithm has been proposed as a new way to analysis and explore mass-cytometry data. This algorithm performs a density-based down-sampling combined with an agglomerative hierarchical clustering. While SPADE offers new opportunities for identifying cell populations, complementary approaches are needed to improve the characterization of identified cell populations. SPADEVizR is an R package designed to better visualize and analyze SPADE clustering results. We extended the original SPADE outputs with techniques such as parallel coordinates, heatmaps, multidimensional scaling, volcano plots or streamgraph representations. Moreover, the proposed statistical methods allow the identification of SPADE clusters with relevant biological behaviors. Significantly abundant clusters or differentially enriched clusters can be identified using SPADEVizR. In addition, the integration of cell cluster behaviors with additional phenotypical variables can be performed.

License GPL-3

Imports data.table,
flowCore,
ggdendro,
ggnetwork,
ggplot2,
gridExtra,
gtools,
igraph,
MASS,
plyr,
reshape2

LazyData true

Suggests knitr

VignetteBuilder knitr

RoxygenNote 5.0.1

R topics documented:

`abundantClustersViewer` *abundantClustersViewer*

Description

Generate a scatter plot representation displaying for each cluster its m

Usage

```
abundantClustersViewer(AC, show.cluster.sizes = TRUE,
  show.all.labels = FALSE)
```

Arguments

`AC`	an object of class AC (object returned by the 'computeAC()' function)
`show.cluster.sizes`	a logicial specifing if points size are related to cell count or not (TRUE by default)
`show.all.labels`	a logicial specifing if all cluster label must be show or just significant cluster

Details

By default, only significant abundant clusters are labeled. Labels for all clusters can be displayed by setting the 'all.label' parameter to TRUE.

Value

a ggplot object

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

Description

The 'AC' object is a S4 object containing the information related to the abundant clusters in a given biological condition.

This object contains all information about the statistical parameters used.

Details

A cluster is considered as a significant abundant cluster if its associated p-value and mean are below the specific thresholds 'th.pvalue' and 'th.mean'.

The 'print()' and 'show()' can be used to display a summury of this object. Moreover all information about this object could be saved as a tab separated file using the 'export()' method. This object is returned by the 'computeAC()' function.

Slots

`sample.names` a character vector containing the samples used to compute the abundant clusters

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

`use.percentages` a logical specifying if computation was performed on percentage of cell abundance

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

`method.adjust` a character containing the name of the multiple correction method used (if any)

`th.mean` a numeric value specifying the mean threshold

`th.pvalue` a numeric value specifying the p-value threshold

`result` a data.frame containing for each cluster (first column): the mean (second column) and the standard deviation (third column) of the biological condition, the associated p-value (fourth column) and a logical (fifth column) specifying if the cluster is significantly abundant.

`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

The 'CC' object is a S4 object containing the information related to the clusters correlated with a phenotypic variable.

This object contains all information about the statistical parameters used.

Details

A cluster is considered as a significant correlated cluster if its associated p-value and correlation threshold are below the specific thresholds 'th.pvalue' and 'th.correlation'.

The 'print()' and 'show()' can be used to display a summury of this object. Moreover all information about this object could be saved as a tab separated file using the 'export()' method. This object is returned by the 'computeCC()' function.

Slots

`sample.names` a character vector containing the samples used to compute correlated clusters

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

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

`use.percentages` a logical specifying if computation was performed on percentage of cell abundance

`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 (if any)

`th.correlation` a numeric value specifying the correlation threshold (R)

`th.pvalue` a numeric value specifying the p-value threshold

`result` a data.frame containing for each cluster (first column): the coefficiant of correlation R (second column) , the associated p-value (third column) and a logical (fourth column) specifying if the cluster is significantly correlated.

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 = 0.7)
```

Arguments

Results	a Results or SPADEResults object
method	a character specifying the clustering method among one of those : 'hierarchical_h','hierarchical_k','kmeans','eigencell','clique' (hierarchical_h by default)
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 = 0.7)
```

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 rownames) 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

xxx

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 p-values should be corrected using multiple correction methods among : "holm", "hochberg", "hommel", "bonferroni", "BH", "BY" and "fdr" (from 'stats::p.adjust' method)
th.pvalue	a numeric specifying the p-value threshold (0.05 by default)
th.mean	a numeric specifying the mean threshold (0 by default)

Value

a AC object

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

Description

xxx

Usage

```
computeCC(Results, variable, use.percentages = TRUE, method = "pearson",
  method.adjust = NULL, th.pvalue = 0.05, th.correlation = 0.75)
```

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 p-values should be corrected using multiple correction methods among : "holm", "hochberg", "hommel", "bonferroni", "BH", "BY" and "fdr" (from 'stats::p.adjust' method)
th.pvalue	a numeric specifying the p-value threshold (0.05 by default)
th.correlation	a numeric specifying the absolute value of the correlation coefficient threshold (0.75 by default)

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

xxx

Usage

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

Arguments

Results	a Results or SPADEResults object
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 p-values should be corrected using multiple correction methods among : "holm", "hochberg", "hommel", "bonferroni", "BH", "BY" and "fdr" (from 'stats::p.adjust' method)
method.paired	a logical indicating if the statistical test must be performed in a paired manner
th.pvalue	a numeric specifying the p-value threshold (0.05 by default)
th.fc	a numeric specifying the fold-change threshold (1 by default)

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 to compute the maker range quantiles.

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

This function performs the exact calculation of quantiles with all cells but needs more memory than computeQuantile.heuristic.

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

Details

This function performs an approximate calculation of quantiles using less memory than compute-Quantile.

Value

a numeric matrix with bounds

correlatedClustersViewer *correlatedClustersViewer*

Usage

```
correlatedClustersViewer(CC, show.cluster.sizes = TRUE,
  show.all.labels = FALSE)
```

Arguments

CC	an object of class CC (object returned by the 'computeCC()' function)
show.cluster.sizes	a logicial specifing if points size are related to cell count or not (TRUE by default)
show.all.labels	a logicial specifing if all cluster label must be show or just significant cluster

Details

By default, only significant correlated clusters are labeled. Labels for all clusters can be displayed by setting the 'all.label' parameter to TRUE.

Value

a ggplot object

CountViewer *CountViewer*

Description

Generate a two dimensional vizualisation showing the number of cells (sum of selected samples) of each cluster.

Usage

```
CountViewer(Results, samples = NULL, clusters = NULL, min.cells = 0,
  sort = TRUE, show.samples = TRUE)
```

Arguments

Results	a SPADEResults or Results object
samples	a named vector providing the correspondence between samples name (in rowname) 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
min.cells	a numeric specifying the minimun number of cell (sum of all selected samples) to display a cluster

`sort` a logical specifying if clusters will be to be sorted (descending) based on the sum of all selected samples for each cluster.

`show.samples` a logical specifying if the number of cells for all selected samples will be displayed.

Details

xxx

Value

a ggplot object

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

Description

The 'DEC' object is a S4 object containing the information related to the differentially enriched clusters between two given biological conditions.

This object contains all information about the statistical parameters used.

Details

A cluster is considered as a differentially enriched cluster if its associated p-value and fold-change are below the specific thresholds 'th.pvalue' and 'th.fc'.

The 'print()' and 'show()' can be used to display a summury of this object. Moreover all information about this object could be saved as a tab separated file using the 'export()' method. This object is returned by the 'computeDEC()' function.

Slots

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

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

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

`use.percentages` a logical specifying if computation was performed on percentage of cell abundance

`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 (if any)

`method.paired` a logical indicating if the statistical test have been performed in a paired manner

`th.fc` a numeric value specifying the fold-change threshold

`th.pvalue` a numeric value specifying the p-value threshold

`result` a data.frame containing for each cluster (first column): the fold-change (second column) and the standard deviation (third column) for the first biological condition, the fold-change (fourth column) and the standard deviation (fifth column) for the second biological condition, the associated p-value (sixth column) and a logical (seventh column) specifying if the cluster is significantly differentially enriched.

dendro *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 *Distogram Viewer*

Description

Generate a distogram representation showing the marker co-expression.

Usage

```
distogramViewer(Results)
```

Arguments

Results	a SPADEResults or Results object

Details

xxx

Value

a list of ggplot objects

EnrichmentProfiles-class

Classification of clusters based on their enrichment profiles (EnrichmentProfiles class) definition

Description

The 'EnrichmentProfiles' is a S4 object containing the information related to the cluster classification based on theirs enrichment profiles.

This object contains all information about the classification method and parameters used.

Details

Five methods are available to classify cellular clusters: 'hierarchical_k', 'hierarchical_h', 'kmeans', 'eigencell' and 'clique'. Each method can paramaterized using the 'method.parameter' parameter.

The 'print()' and 'show()' can be used to display a summury of this object. Moreover all information about this object could be saved as a tab separated file using the 'export()' method. This object is returned by the 'classifyEnrichmentProfiles()' function.

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 (case insentive)
`colnames.FCS`	a character vector containing colnames if data is a FCS flowset

Details

If the data parameter is a dataframe the colnames.FCS parameter is ignored but if the data parameter is a flowset, the colnames.FCS parameter is required.

Value

a numeric matrix without the cell markers to exclude

`export` *Exportation of SPADEVizR objects*

Description

Exports a SPADEVizR object into a tab separated file.

Usage

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

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

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

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

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

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

## S4 method for signature 'EnrichmentProfiles'
export(object, filename = "export.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 removed.

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)

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

Description

This function is used internally to build the element needed for an heatmap

Usage

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

Arguments

`matrix`
: 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 *Internal - ggheatmap.plot*

Description

ggheatmap.plot display the heatmap build by ggheatmap

Usage

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

Arguments

`list`
: 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` *Internal - Extraction of ggplot axes*

Description

This function is used internally to extract axes from a 'ggplot' objet.

Usage

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

Arguments

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

Details

It is to note that x and y are mutuality excluded (both cannot be both TRUE) with priority to x.

Value

an object of class xxx

`g_legend` *Internal - Extraction of ggplot legend*

Description

This function is used internally to extract the legend from a 'ggplot' objet.

Usage

```
g_legend(gplot)
```

Arguments

`a.gplot`	a ggplot2 plot

Value

a ggplot2 legend

importResults *Import clustering results generated by other algorithms*

Description

The 'importX()' function imports cell clustering results from two dataframes ('cells.count' and 'marker.expressions'). This function returns a 'Result' object.

Usage

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

Arguments

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

Details

The detailled format of 'cells.count' dataframe must be formated as following:

cluster	sample1	sample1
cluster1	749	5421
cluster2	450	412

The detailled format of 'marker.expressions' dataframe must be formated as following:

sample	cluster	marker1	marker2
sample1	cluster1	0.2	0.3
sample1	cluster2	0.1	0.3
sample2	cluster1	0.5	2.3
sample2	cluster2	1	1.3

Value

Results a result object

importSPADEResults *Import clustering results generated by SPADE*

Description

The 'importSPADEResults()' function imports SPADE cell clustering results from a specified path. This function returns a 'SPADEResult' object.

This function import the expression matrix and count matrix as well as the SPADE tree. This function apply an hyperbolic sine transformation to imported FCS data and compute the maker range quantiles.

Usage

```
importSPADEResults(path, dict = data.frame(),
  exclude.markers = c("cell_length", "FileNum", "density", "time"),
  probs = c(0.05, 0.95), use.raw.medians = FALSE,
  quantile.heuristic = FALSE)
```

Arguments

path	a character specify the path of SPADE results folder
dict	a two column dataframe providing the correspondance between the original marker names (first column) and the real marker names (second column)
exclude.markers	a character vector of markers to exclude (case insentive)
probs	a vector of probabilities with 2 values in [0,1] to compute maker range quantiles. First is the lower bound and second is the upper bound.
use.raw.medians	a logicial specifying if "transformed" or "raw" medians will be use in the cluster expression matrix (FALSE by default)
quantile.heuristic	a logicial specifying if maker range quantiles are computed using all cells (FALSE), or is the means of the quantile of each samples (TRUE)

Details

The computation of maker range quantiles can be approximated using quantile.heuristic parameter which is more efficient in term of loading time and memory usage.

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 *Class name*

Description

Provides the name of each SPADEVizR 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 SPADEVizR 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

a character specifying the name of the object

none

none

none

none

none

none

`PhenoProfiles-class` *Classification of clusters based on their phenotype profiles (PhenoProfiles class) definition*

Description

The 'PhenoProfiles' is a S4 object containing the information related to the cluster classification based on theirs marker expressions.

This object contains all information about the classification method and parameters used.

Details

Five methods are available to classify cellular clusters: 'hierarchical_k', 'hierarchical_h', 'kmeans', 'eigencell' and 'clique'. Each method can paramaterized using the 'method.parameter' parameter.

The 'print()' and 'show()' can be used to display a summury of this object. Moreover all information about this object could be saved as a tab separated file using the 'export()' method. This object is returned by the 'classifyPhenoProfiles()' function.

Slots

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

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

`method.parameter` a named list of parameters used by the classification method

`cluster.size` a numeric vector containing the number of cells associated with each cluster (– sum of all samples –)

`class.number` a numeric value specifying the number of clusters

`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

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

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

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

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

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

Arguments

xxx

Details

xxx

Value

xxx

---

print *Textual previews for all SPADEVizR objects*

Description

Prints a previews for a SPADEVizR object.

Prints a previews for a SPADEResults 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 SPADEVizR object
x	a SPADEResults object
x	a EnrichmentProfiles object

Value

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

The Results object is a S4 object containing cell clustering results obtained from various automatic gating algorithms.

This object mainly stores the count matrix (i.e. the number of cells associated with each cluster of each sample) and the cell cluster phenotypes (i.e. the marker median expressions for each cluster). It is to note that the Results object is a super class of the SPADEResult object.

Details

The 'cells.count' dataframe stores the number of cells associated with each cluster of each sample. This dataframe has in row the clusters and in column the samples.

The 'marker.expressions' dataframe stores the marker median expressions for each cluster. This dataframe has in the first the sample names, in the second column the cluster names, and the maker median expressions in the others columns.

The 'print()' and 'show()' can be used to display a summury of this object. Moreover all information about this object could be saved as a tab separated file using the 'export()' method. This object is returned by the 'importX()' function.

Slots

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

`marker.expressions` 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 specifying the number of cell clusters

---

show *Textual previews for SPADEVizR objects*

Description

Show a previews for a SPADEVizR objects.

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 SPADEVizR 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

SPADEResults-class *SPADEResults class definition*

Description

The 'SPADEResults' object is a S4 object containing cell clustering results obtained from SPADE.

This object inherits from the 'Result' object and stores the count matrix (i.e. the number of cells associated with each cluster of each sample) and the cell cluster phenotypes (i.e. the marker median expressions for each cluster). In addition to the 'Result' object, the 'SPADEResults' object contains information about SPADE clustering results, such as the SPADE tree, the clustering makers and the FCS files.

Details

The 'print()' and 'show()' can be used to display a summury of this object. Moreover all information about this object could be saved as a tab separated file using the 'export()' method. This object is returned by the 'importSPADEResult()' function.

Slots

`use.raw.medians` a logical specifying if the marker expressions correspond to the raw or transformed data

`dictionary` a two column data.frame providing the correspondence between the original marker names (first column) and the real 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 numeric data.frame containing the quantiles for each each markers of cluster

`graph` a igraph object containing the SPADE tree

`graph.layout` a numeric 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

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, show.cluster.sizes = TRUE,
  show.all.labels = FALSE)
```

Arguments

`DEC`	an object of class DEC (object returned by the 'computeDEC()' function)
`fc.log2`	a logicial specifing if fold-change or log2(fold-change) is use
`show.cluster.sizes`	a logicial specifing if points size are related to cell count or not (TRUE by default)
`show.all.labels`	a logicial specifing if all cluster labels must be show or just significant cluster
`SPADEResults`	the SPADEViewer result object

Details

By default, only significant differentially enriched clusters are labeled. Labels for all clusters can be displayed by setting the 'all.label' parameter to TRUE.

Value

a ggplot object

Index