Package 'SPADEVizR'

May 19, 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. Automatic gating algorithms can be used to automatically identified clusters of cells having similar phenotypes. 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. XXX 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. This package extends the original SPADE outputs with techniques such as parallel coordinates, heatmaps, multidimensional scaling, volcano plots or streamgraph representations. Moreover several statistical methods allow the identification of SPADE clusters with relevant biological behaviors.

License GPL-3 | file LICENCE

```
Imports data.table,
      flowCore,
      ggdendro,
      ggnetwork,
      ggplot2,
      ggrepel,
      grid,
      gridExtra,
      gtable,
      gtools,
      grDevices,
      igraph,
      MASS,
      methods,
      network,
      plyr,
      reshape2
```

Suggests knitr VignetteBuilder knitr RoxygenNote 5.0.1

${\sf R}$ topics documented:

abundantClustersViewer	3
AC-class	4
biplotViewer	
boxplotViewer	
CC-class	6
classifyEnrichmentProfiles	7
classifyPhenoProfiles	8
clusterViewer	9
computeClique	10
computeEigenCellClusters	10
computeHierarchicalClustering	11
computeKmeans	12
computePhenoTable	12
computeQuantile	13
computeQuantile.approximation	13
correlatedClustersViewer	14
countViewer	14
DEC-class	15
distogramViewer	16
EnrichmentProfiles-class	17
exclude.markers	17
export	18
filter.medians	19
generateReport	19
ggheatmap	20
ggheatmap.plot	
g_axis	
g_dendro	22
g_legend	23
identifyAC	23
identifyCC	24
identifyDEC	24
importResults	
importSPADEResults	
kineticsViewer	
load.flowSet	
MDSViewer	28
names	29
PhenoProfiles-class	30
phenoViewer	30
- plot	31
rint	32
profiles Viewer	32
rename.markers	33
Results-class	34

abundantClustersViewer 3

Index	3	39
	volcanoViewer	,7
	unload.flowSet	57
	treeViewer	6
	streamgraphViewer	6
	SPADEResults-class	5
	show	

abundantClustersViewer

Visualization of abundant clusters

Description

Generates a scatter plot representation showing for each cluster its mean abundance and associated p-value.

Usage

```
abundantClustersViewer(AC, show.cluster_sizes = TRUE,
    show.all_labels = FALSE)
```

Arguments

Details

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

Value

```
a 'ggplot' object
```

4 biplot Viewer

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. Moreover this object contains all parameters used in the statistical analysis.

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 'identifyAC()' 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.

biplotViewer

biplotViewer

Description

Generates a biplot representation with two markers

```
biplotViewer(SPADEResults, x.marker, y.marker, samples = NULL,
   clusters = NULL, sample.merge = FALSE, resample.ratio = NULL)
```

boxplotViewer 5

Arguments

SPADEResults	a SPADEResults object (Results object is not accepted)
x.marker	a character indicating the marker name of the first dimension
y.marker	a character indicating the marker name of the second dimension
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 names to be visualized (by default all clusters will be used)
sample.merge	a logical specifying if the selected samples must be merged in a single biplot
resample.ratio	a numeric ratio (between 0 and 1) specifying the downsample ratio to show less dots (or NULL)

Details

In such representation, each dot corresponds to a cell profile and dots are ploted in a 2-dimentional space corresponding to the selected markers. When too cells dots are displayed, it can require some seconds. In order to seep up the computation, it is possible to reduce the number of cells displayed (downsampling) using the 'resample.ratio' parameter.

Value

a 'ggplot' object

boxplotViewer	Visualization of cluster enrichment profiles conditions	

Description

Generate a boxplot representation displaying the cell abundance for each cluster. Clusters are gathered by given biological conditions.

Usage

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

Arguments

Results a SPADEResults or Results object		
conditions conditions a named vector providing the correspondence between a sam (in row names) and the condition of this sample or NA to exclude		
clusters a character vector containing the clusters names to be visualized (by declusters will be displayed)		
use.percentages	3	
	a logical specifying if the visualization must be performed on percentage	
show.legend	a logical specifying if the legend must be displayed	
show.violin	a logical specifying if the count distribution must be displayed	

6 CC-class

Details

Cells clusters are colored based on theirs associated biological samples.

Value

a 'ggplot' object

CC-class

Correlated Clusters (CC class) definition

Description

The 'CC' object is a S4 object containing coefficient of correlation associated between each cluster and a phenotypic variable. Moreover this object contains all parameters used in the statistical analysis.

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 'identifyCC()' 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

Classification of clusters based on their enrichment profiles

Description

Classifies clusters based on their enrichment profiles.

Usage

```
classifyEnrichmentProfiles(Results, method = "hierarchical_h",
  method.parameter = NULL)
```

Arguments

Results a Results or SPADEResults object

method a character specifying the clustering method among one of those: 'hierarchical_h','hierarchical_k','k-

means', 'eigencell', 'clique' (hierarchical_h by default)

method.parameter

a numeric specifying the numeric value required by the selected method

Details

The classification is done on cell abundances of each clusters and could be performed using 5 methods:

- "hierarchical_k" This method first compute the Pearson correlation matrix and then use this matrix to performs a hierarchical classification. The hierarchical classification is cutted in order to return the desired number of classes. This number of classes must be provided as a numeric integer using the 'method.parameter' parameter. It is to note that negative correlations are considered as uncorrelated
- "hierarchical_h" (default method) This method works in the same way than 'hierarchical_k' but the height where the hierarchical tree is specified. This height is a correlation threshold (a numeric double between 0 and 1 included, default is 0.7) provided using the 'method.parameter' parameter.
- "kmeans" This method works as described in the R stats documentation (?kmeans) using the 'method.parameter' parameter to specify the desired number of classes.
- "eigencell" This method performs an eigen vector decomposition and then calculate the correlations between cluster enrichment profiles and these vectors. Clusters which correlate above a specific threshold with the same eigen vector are classified together. This correlation threshold (a numeric double between 0 and 1 included, default is 0.8) provided using the 'method.parameter' parameter.
- "clique" This method first compute the Pearson correlation matrix and then use this matrix to generate an undirected graph. In this graph, an edge is drawn between two nodes if the correlation coefficient in the adjacency matrix is above a specific threshold. This correlation threshold (a numeric double between 0 and 1 included, default is 0.7) provided using the 'method parameter' parameter. After building the graph, the method looking for the largest cliques wich are considered as classes of nodes. Cliques correspond to subgraph in which every two distinct vertices are adjacent.

8 classifyPhenoProfiles

Value

a S4 object of class 'EnrichmentProfiles'

classifyPhenoProfiles Classification of clusters based on their phenotype profiles

Description

Classifies clusters based on their phenotype profiles.

Usage

```
classifyPhenoProfiles(Results, method = "hierarchical_h",
  method.parameter = NULL)
```

Arguments

Results a Results or SPADEResults object

method a character specifying the clustering method among one of those: "hierarchi-

cal_h", "hierarchical_k", "k-means", "eigencell", "clique"

method.parameter

a numeric specifying the numeric value required by the selected method

Details

The classification is done on cell abundances of each clusters and could be performed using 5 methods:

- "hierarchical_k" This method first compute the Pearson correlation matrix and then use this
 matrix to performs a hierarchical classification. The hierarchical classification is cutted in
 order to return the desired number of classes. This number of classes must be provided as a
 numeric integer using the 'method.parameter' parameter. It is to note that negative correlations
 are considered as uncorrelated
- "hierarchical_h" (default method) This method works in the same way than 'hierarchical_k' but the height where the hierarchical tree is specified. This height is a correlation threshold (a numeric double between 0 and 1 included, default is 0.7) provided using the 'method.parameter' parameter.
- "kmeans" This method works as described in the R stats documentation (?kmeans) using the 'method.parameter' parameter to specify the desired number of classes.
- "eigencell" This method performs an eigen vector decomposition and then calculate the correlations between cluster expressions and these vectors. Clusters which correlate above a specific threshold with the same eigen vector are classified together. This correlation threshold (a numeric double between 0 and 1 included, default is 0.8) provided using the 'method.parameter' parameter.
- "clique" This method first compute the Pearson correlation matrix and then use this matrix to generate an undirected graph. In this graph, an edge is drawn between two nodes if the correlation coefficient in the adjacency matrix is above a specific threshold. This correlation threshold (a numeric double between 0 and 1 included, default is 0.7) provided using the 'method parameter' parameter. After building the graph, the method looking for the largest cliques wich are considered as classes of nodes. Cliques correspond to subgraph in which every two distinct vertices are adjacent.

clusterViewer 9

Value

a S4 object of class 'PhenoProfiles'

|--|

Description

Generates a parallel coordinate plot representation showing for each cluster the marker median expressions.

Usage

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

Arguments

Results	a SPADEResults or Result object
clusters	a character vector containing the clusters names to be visualized (by default all clusters will be displayed)
samples	named vector providing the correspondence between samples name (in rowname) and the logical value TRUE to use these samples (all samples by default)
markers	a character vector specifying the markers to be displayed
show.mean	a character specifying if marker means expression should be displayed, possible value are among : "none", "only" or "both"

Details

If 'Results' is an object of class 'SPADEResults' then the quantile ranges will be displayed using a ribbon

The 'samples' parameter required a named vector providing the correspondence between a sample name (in rowname) and the logical value TRUE to show this sample or FALSE otherwise.

The 'show.mean' parameter allows to visualize three kinds of information:

- "none" value will show marker median expressions for each selected samples;
- "only" value will show only the mean of median maker expressions for all selected samples (displayed as black dashed line);
- "both" value will show marker median expressions for each selected samples together with the mean of median maker expressions for all selected samples.

Value

```
a list of 'ggplot' objects
```

computeClique

Internal - Clique percolation classification

Description

This function is used internally to classify clusters enrichment profiles or phenotype profiles using a clique percolation algorithm.

Usage

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

Arguments

```
data a numeric matrix with all clusters in rownames clique.correlation.th
a numeric value indicating the correlation coefficient threshold
```

Details

This method first compute the Pearson correlation matrix and then use this matrix to generate an undirected graph. In this graph, an edge is drawn between two nodes if the correlation coefficient in the adjacency matrix is above a specific threshold. This correlation threshold (a numeric double between 0 and 1 included, default is 0.7) provided using the 'clique.correlation.th' parameter. After building the graph, the method looking for the largest cliques wich are considered as classes of nodes. Cliques correspond to subgraph in which every two distinct vertices are adjacent.

Value

a dataframe containing for each cluster, its name and class

```
computeEigenCellClusters
```

Internal - Eigen vector classification

Description

This function is used internally to classify clusters enrichment profiles or phenotype profiles using eigen vector decomposition.

Usage

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

Arguments

```
data a numeric matrix with all clusters in rownames eigencell.correlation.th a numeric value indicating the correlation coefficient threshold
```

Details

This method compute the performs a eigen vector decomposition and then calculate the correlations between the matrix rows and these vectors. Clusters which correlate above a specific threshold with the same eigen vector are classified together. This correlation threshold (a numeric double between 0 and 1 included, default is 0.8) provided using the 'eigencell.correlation.th' parameter.

Value

a dataframe containing for each cluster, its name and class

computeHierarchicalClustering

Internal - Hierarchical classification

Description

This function is used internally to classify clusters enrichment profiles or phenotype profiles using a hierarchical algorithm.

Usage

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

Arguments

data a matrix with all clusters in rownames
class.number a numeric specifying the number of classes
hierarchical.correlation.th
a numeric value specifying the cut height

Details

This function compute the Pearson correlation matrix associated to the provided matrix. It is to note that negative correlations are considered as uncorrelated. This correlation matrix is used to performs a hierarchical classification. If 'class.number' parameter is NULL, classification will be determined based on the cut height correlation threshold (i.e. 'hierarchical.correlation.th' parameter)

Value

a dataframe containing for each cluster, its name and class

12 computePhenoTable

computeKmeans

Internal - Kmeans classification

Description

This function is used internally to classify clusters enrichment profilies or phenotype profiles using a k-means algorithm.

Usage

```
computeKmeans(data, k = NULL)
```

Arguments

data a numeric matrix with cluster names in rownames k a numeric specifying the desired number of classes

Details

This method works as described in the R stats documentation (?kmeans) using the 'k' parameter to specify the desired number of classes.

Value

a dataframe containing for each cluster, its name and class

 ${\tt computePhenoTable}$

Internal - Generatate marker expression scores describing phenotypes

Description

This function is used internally to generate a melted numeric matrix of discrete expression scores for each marker of each cluster.

Usage

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

Arguments

SPADEResults a SPADEResults object

num a numeric value specifying the number of markers expression categories

Details

NA values are removed

Value

a numeric matrix of expression scores

computeQuantile 13

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 numeric vector of 2 values specifying the quantiles to compute

Details

This function performs the exact calculation of quantiles with all cells but needs more ressources (time and memory usage) than 'computeQuantile.approximation'.

Value

a numeric matrix containing the quantiles of each marker

```
computeQuantile.approximation
```

Internal - Compute quantile with FCS flowset sample by sample

Description

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

Usage

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

Arguments

flowset a flowCore flowset

probs a numeric vector of 2 values specifying the quantiles to compute

Details

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

Value

a numeric matrix containing the quantiles of each marker

14 countViewer

correlatedClustersViewer

Visualization of correlated clusters

Description

Generate a scatter plot representation showing for each cluster

Usage

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

Arguments

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

Visualization of cluster sizes

Description

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

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

DEC-class 15

Arguments

Results	a SPADEResults or Results object
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 names to be visualized (by default all clusters will be displayed)
min.cells	a numeric specifying the minimum 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

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. Moreover this object contains all parameters used in the statistical analysis.

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 'identifyDEC()' 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

16 distogramViewer

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.

distogramViewer

Visualization of marker co-expressions

Description

Generate a distogram representation showing the marker co-expressions.

a CDA DEPagults or Pagults object

Usage

```
distogramViewer(Results, clusters = NULL, samples = NULL, markers = NULL)
```

Arguments

Docul+0

Results	a SPADERESURS OF RESURS ODJECT
clusters	a character vector containing the clusters names to be use (by default all clusters will be used)
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)

markers a character vector specifying the markers to be displayed

Details

A Pearson correlation matrix is calculated between selected markers using selected clusters and samples. High positive correlated markers are shown by a green tile at their perpendicular intersection. In the same way, absence of correlation are shown by black tiles and negative correlation by red tiles.

Value

a list of 'ggplot' objects

EnrichmentProfiles-class 17

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 parameterized using the 'method.parameter' parameter.

The 'print()' and 'show()' can be used to display a summary 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

class.number a numeric providing the number of classes

method a character specifying the method used to classify cluster

method.parameter a numeric parameters associated with the chosen method

classes a two column dataframe with the cluster in first column and corresponding class in the

second column

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 intensive)

colnames . FCS a character vector containing column names 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.

18 export

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 19

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 logical specifying if "transformed" or "raw" medians will be use (FALSE by default)

Value

a numeric matrix without the cell markers to exclude

generateReport

Generate a report including SPADEVizR plots.

Description

Generate a customizable PDF report based on SPADEVizR vizualisation features. Available plots are :

- "count" (included by default):Display an representation showing the number of cells for each cluster
- "tree" (included by default):Display a tree representation showing combined SPADE trees
- "pheno" (included by default):Display an heatmap representation
- "boxplot":Display a boxplot representation. This plot required to provide the 'conditions' parameter
- "kinetics":Display a kinetic representation for each cluster. This plot required to provide the 'assignments' parameter
- "stream":Display a streamgraphViewer representation showing the evolution of cells abundance
- "cluster" (included by default):Display a parallel coordinate representation showing for each cluster the marker median expression
- "MDSclusters" (included by default):Display the cluster similarities using MDS
- "MDSsamples":Display the samples similarities using MDS. This plot required to provide the 'assignments' parameter
- "disto" (included by default): Display a distogram representation showing the marker co-expressions

20 ggheatmap

• "kinetics_cluster":Display a kinetic representation and a parallel coordinate juxtaposed (are arranged one on the side of the other) for each cluster

• "boxplot_cluster":Display a boxplot representation and a parallel coordinate juxtaposed (are arranged one on the side of the other) for each cluster

Usage

```
generateReport(Results, PDFfile = "report.pdf", plot.names = c("count",
   "pheno", "tree", "disto", "MDSclusters", "cluster"), clusters = NULL,
   markers = NULL, assignments = NULL, conditions = NULL,
   stat.objects = list(), profile.objects = list(), width = 29.7,
   height = 21)
```

Arguments

Results	a 'SPADEResults' or 'Result' object
PDFfile	a character specifying the output path
plot.names	a character vector specifying the names (see details) and the order of the desired plots
clusters	a character vector of clusters to include in the report (all will be included by default)
markers	a character vector of markers to include in the report (all will be included by default)
assignments	a 2 column data.frame with the samples names in row names providing firstly the time-points (numeric) and secondly the individuals (character) of the experiment
conditions	conditions a named vector providing the correspondence between a sample name (in row names) and the condition of this sample or NA to exclude
stat.objects	a vector of stat.object to be displayed in the report (object of class 'DEC', 'AC' or 'CC')
profile.object	s
	a vector of profile.objects to be displayed in the report (object of class 'Pheno-Profiles' or 'EnrichmentProfiles')
width	a numeric specifying the plot width in centimeter
height	a numeric specifying the plot height in centimeter

ggheatmap	Internal -	Create a list of	elements al	lowing to build a	i heatmap
-----------	------------	------------------	-------------	-------------------	-----------

Description

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

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

ggheatmap.plot 21

Arguments

matrix a numeric matrix containing the markers expression categories

dendrogram.type

a character specifying the look of dendrograms ("rectangle" or "triangle", "rect-

angle" by default)

num a numeric value specifying the number of markers expression categories

clustering.markers

a character vector of clustering markers

Value

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

ggheatmap.plot

Internal - Generate an heatmap by assembling elements

Description

This function displays the heatmap elements 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

Value

a ggplot2 axis

g_axis

Internal - Extraction of ggplot axes

Description

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

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

g_dendro

Arguments

```
gplot a 'ggplot' plot

x.axis a logical value specifying if the x-axis must be extract

y.axis a logical value specifying if the y-axis must be extract
```

Details

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

Value

```
a 'ggplot' axis object
```

g	dendro	
~-	_acriar o	

Internal - Build a dendrograms plot

Description

This function is used internally to generate a 'ggplot' dendrogram.

Usage

```
g_dendro(dist, row = !col, col = !row)
```

Arguments

dist	a numeric matrix containing distances between objects
row	a logical value specifying if the horizontal dendrogram must be computed
col	a logical value specifying if the vertical dendrogram must be computed

Details

It is to note that 'row' and 'col' are mutuality excluded (both cannot be both TRUE) with priority to row.

Value

a 'ggplot' dendrogram object

g_legend 23

g_legend	Internal - Extraction of ggplot legend

Description

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

Usage

```
g_legend(gplot)
```

Arguments

gplot a 'ggplot' plot

Value

a 'ggplot' legend object

identifyAC

Identification of the Abundant Clusters

Description

This function is used to identify the abundant clusters. That is to say clusters that have cell abundance statistically greater than a specific threshold.

Usage

```
identifyAC(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 row

names) and the logical value TRUE to test abundance for this sample or FALSE $\,$

otherwise

use.percentages

a logical specifying if the computations should be performed on percentage

method a character specifying the statistical method used to identify the abundant clus-

ters. 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 cor-

rection methods among: "holm", "hochberg", "hommel", "bonferroni", "BH",

"BY" and "fdr" (from 'stats::p.adjust' method)

th.pvalue a numeric specifying the p-value threshold

th.mean a numeric specifying the abundance mean threshold

Value

```
a S4 object of class 'AC'
```

24 identifyDEC

			•	_	\sim	_
٦.	de	nt	٦.	+١	/(•
	uc	ווע	_	ι ۱	/ 🔾	v.

Identification of the correlation of SPADE cluster with a phenotype

Description

This function is used to identify correlated clusters. That is to say clusters that correlate with a phenotypic variable.

Usage

```
identifyCC(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 rownames) 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 cor-

rection methods among: "holm", "hochberg", "hommel", "bonferroni", "BH",

"BY" and "fdr" (from 'stats::p.adjust' method)

th.pvalue a numeric specifying the p-value threshold

th.correlation a numeric specifying the absolute value of the correlation coefficient threshold

Value

a S4 object of class 'CC'

identifyDEC

Identification of the Differentially Enriched Clusters

Description

This function is used to identify differentially enriched clusters. That is to say clusters that are differentially abundant between two biologicals conditions.

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

importResults 25

Arguments

Results a 'Results' or 'SPADEResults' object

conditions a named vector providing the correspondence between a sample name (in row

names) and the condition of this sample : NA to exclude a sample from tests, $\boldsymbol{1}$

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 cor-

rection 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
th.fc a numeric specifying the fold-change threshold

Value

a S4 object of class 'DEC'

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 abundances 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 'cells.count' dataframe must be formated as following:

cluster	sample1	sample
cluster1	749	5421
cluster2	450	412

26 importSPADEResults

The '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

a S4 object of class 'Results'

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

Arguments

path a character specify the path of SPADE results folder

dictionary a two column dataframe providing the correspondence between the original

marker names (first column) and the real marker names (second column)

exclude.markers

a character vector of markers to exclude (case insensitive)

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 logical specifying if arcsinh transformed or raw medians will be used in the

cluster expression matrix (FALSE by default)

quantile.approximation

a logical 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.approximation' parameter which is more efficient in term of loading time and memory usage.

kinetics Viewer 27

Value

a S4 object of class 'SPADEResults'

kineticsViewer Visualization of cluster enrichment profiles kinetics

Description

Generates a kinetics plot representation showing for each cluster its enrichment profiles at each time-point of each individual.

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 row names providing firstly the

time-points (numeric) and secondly the individuals (character) of the experiment

clusters a character vector containing the clusters names to be visualized (by default all

clusters will be displayed)

use.percentages

a logical specifying if the visualization should be performed on percentage

Details

XXX

Value

a 'ggplot' object

load.flowSet

Load FCS files object into a 'SPADEResult' object

Description

This function loads the FCS files to the 'flowset' slot of the 'SPADEResult' object.

```
load.flowSet(SPADEResult = NULL, fcs.files, dictionary, exclude.markers,
  use.raw.medians)
```

28 MDSViewer

Arguments

SPADEResult a SPADEResult object (optional)

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

dictionary a two column data frame providing the correspondence between the original

marker names (first column) and the real marker names (second column)

exclude.markers

a character vector of markers to exclude (case insensitive)

use.raw.medians

a logical specifying if the arcsinh transformation must be performed or not

Details

If a 'SPADEResult' object is provided, others parameters ('fcs.files', 'dictionary', 'exclude.markers', 'use.raw.medians') will be ignored.

Value

a S4 'flowSet' object

MDSViewer

Visualization of SPADE cluster or sample similarities using MDS

Description

Generate a Multidimensional Scaling (MDS) representation showing the similarities between SPADE results based on theirs enrichment.

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 visualization should be performed on percentage

 $assignments \qquad a \, 2 \, column \, data. frame \, with \, the \, samples \, names \, in \, row \, names \, providing \, firstly \, the \, data \, dat$

time-points (numeric) and secondly the individuals (character) of the experiment

clusters a character vector containing the clusters names to be visualized (by default all

clusters will be displayed)

space a character specifying the space ("clusters" or "samples", "cluster" by default)

dist.method a character string containing the name of the distance measure to use

Details

The 'space' parameter specifying if the cluster or sample similarities will be determined using MDS. When the 'space' parameter is equal to 'sample', the 'assignements' parameter is required. Available method for the 'dist.method' parameter are: "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski"

names 29

Value

```
a list of 'ggplot' objects
```

names

Definition of class names

Description

Provides the name of each SPADEVizR 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

Value

a character providing the name of the object

30 phenoViewer

PhenoProfiles-class Classification of clusters based on their phenotype profiles (Pheno-Profiles 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 parameterized using the 'method parameter' parameter.

The 'print()' and 'show()' can be used to display a summary 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

class.number a numeric value specifying the number of clusters
method a character specifying the method used to classify cluster
method.parameter a named list of parameters used by the classification method

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

phenoViewer

Visualization of all clusters phenotypes as an heatmap

Description

Generates an heatmap representation showing for all clusters the marker median expressions.

Usage

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

Arguments

SPADEResults object (Results object is not accepted)

num a numeric value specifying the number of markers expression categories to use

Details

For each marker, median expressions are discretized in severals categories corresponding to the heat intensities. This number of categories is provided using 'num' parameter.

By default the 'pheno.table' parameter is null and will be automatically compute using the 'computePhenoTable' function.

plot 31

Value

```
a list of 'ggplot' objects
```

plot

Graphical representation for some SPADEVizR objects

Description

This function generates a graphical representation for 'AC', 'DEC', 'CC', 'PhenoProfiles' and 'EnrichmentProfiles' objects.

Usage

```
plot(x, y = NULL, ...)
## S4 method for signature 'DEC,missing'
plot(x, y = NULL, ...)
## S4 method for signature 'AC,missing'
plot(x, y = NULL, ...)
## S4 method for signature 'CC,missing'
plot(x, y = NULL, ...)
## S4 method for signature 'PhenoProfiles,missing'
plot(x, y = NULL, ...)
## S4 method for signature 'EnrichmentProfiles,missing'
plot(x, y = NULL, ...)
```

Arguments

```
x a 'AC', 'DEC', 'CC', 'PhenoProfiles' or 'EnrichmentProfiles' object
y a supplementary parameter transmited to 'AC', 'DEC' or 'CC' object
... some supplementaries parameters transmited to 'AC', 'DEC' or 'CC' object
```

Value

```
a 'ggplot' object
```

32 profiles Viewer

print

Textual previews for all SPADEVizR objects

Description

Prints a previews for a SPADEVizR 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

Χ

a SPADEVizR object

Value

none

 $\verb|profilesViewer|$

profiles Viewer

Description

Generate a graph representation of PhenoProfiles classes

```
profilesViewer(profile.object)
```

rename.markers 33

Arguments

```
profile.object a PhenoProfiles object or an EnrichmentProfiles object
```

Details

XXX

Value

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

34 show

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 summary 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 object.

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

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

## S4 method for signature 'AC'
```

SPADEResults-class 35

```
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

object a SPADEVizR object

Value

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

flowset a flowSet object containing the imported SPADE FCS file

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

36 tree Viewer

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

streamgraphViewer

Visualization of cluster abundance dynamics

Description

Generate a streamgraph representation showing the dynamic evolution of the number of cells in clusters across samples.

Usage

```
streamgraphViewer(Results, order = NULL, clusters = NULL,
  use.relative = FALSE)
```

Arguments

Results a SPADEResults or Results object

order a named vector providing the correspondence between a sample name (in row-

names) and an integer ordering samples (NA to exclude a sample)

clusters a character vector containing the clusters names to be vizualised (by default all

clusters will be displayed)

use.relative a logical specifying if the visualization should be performed on relative abun-

dance

Value

a 'ggplot' object

treeViewer

Visualization of combined SPADE trees

Description

Generates a tree representation showing combined SPADE trees.

Usage

```
treeViewer(SPADEResults, samples = NULL, stat.object = NULL,
marker = NULL)
```

Arguments

SPADEResults object (Results object is not accepted)

samples a named vector providing the correspondence between samples name (in row-

name) and the logical value TRUE to use these samples (all samples by default)

stat.object an AC, DEC or CC object to highligth identified significant clusters in the

SPADE tree

marker a character specifying the marker name to display

unload.flowSet 37

Details

The size of tree nodes are related to the number of cells in each cluster. If the 'stat.object' parameter is provided node outlines are colored according to clusters signifiance. If the 'marker' parameter is provided, the nodes are colored according to mean expression for the selected marker using selected samples.

Value

```
a list of 'ggplot' objects
```

unload.flowSet

Unload 'flowSet' object from a 'SPADEResult' object

Description

This function unloads the 'flowSet' object in a 'SPADEResult' object.

Usage

```
unload.flowSet(SPADEResult)
```

Arguments

```
SPADEResult a SPADEResult object (optional)
```

Value

The new 'SPADEResult' object

volcanoViewer

Visualization of differentially enriched clusters

Description

Generates a Volcano plot representation showing for each cluster

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 logical specifying if fold-change or log2(fold-change) is use

show.cluster.sizes

a logical specifying if dot sizes are proportional to cell counts

show.all_labels

a logical specifying if all cluster labels must be show or just significant cluster
```

38 volcano Viewer

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

abundantClustersViewer, 3	ggheatmap, 20
AC (AC-class), 4	ggheatmap.plot, 21
AC-class, 4	
	identifyAC, 23
biplotViewer, 4	identifyCC, 24
boxplotViewer, 5	identifyDEC, 24
	<pre>importResults, 25</pre>
CC (CC-class), 6	importSPADEResults, 26
CC-class, 6	
classifyEnrichmentProfiles, 7	kineticsViewer, 27
classifyPhenoProfiles, 8	
clusterViewer, 9	load.flowSet, 27
computeClique, 10	
computeEigenCellClusters, 10	MDSViewer, 28
computeHierarchicalClustering, 11	
computeKmeans, 12	names, 29
computePhenoTable, 12	names, AC-method (names), 29
computeQuantile, 13	names, CC-method (names), 29
computeQuantile.approximation, 13	names, DEC-method (names), 29
correlatedClustersViewer, 14	names,EnrichmentProfiles-method
countViewer, 14	(names), 29
Codific viewer, 14	names, PhenoProfiles-method (names), 29
DEC (DEC-class), 15	names, Results-method (names), 29
DEC-class, 15	names, SPADEResults-method (names), 29
distogramViewer, 16	
distogramviewer, 10	PhenoProfiles (PhenoProfiles-class), 30
EnrichmentProfiles	PhenoProfiles-class, 30
(EnrichmentProfiles-class), 17	phenoViewer, 30
EnrichmentProfiles-class, 17	plot, 31
exclude.markers, 17	plot, AC, missing-method (plot), 31
export, 18	plot,CC,missing-method(plot), 31
export, AC-method (export), 18	plot, DEC, missing-method (plot), 31
export, CC-method (export), 18	plot,EnrichmentProfiles,missing-method
	(plot), 31
export, DEC-method (export), 18	plot,PhenoProfiles,missing-method
export, EnrichmentProfiles-method	(plot), 31
(export), 18	print, 32
export, PhenoProfiles-method (export), 18	print, AC-method (print), 32
export, Results-method (export), 18	print, CC-method (print), 32
0.7	print, DEC-method (print), 32
filter.medians, 19	print, EnrichmentProfiles-method
marks 21	· · · · · · · · · · · · · · · · · · ·
g_axis, 21	(print), 32
g_dendro, 22	print, PhenoProfiles-method (print), 32
g_legend, 23	print, Results-method (print), 32
generateReport, 19	print, SPADEResults-method (print), 32

40 INDEX

```
profilesViewer, 32
rename.markers, 33
Results (Results-class), 34
Results-class, 34
show, 34
show, AC-method (show), 34
show, CC-method (show), 34
show, DEC-method (show), 34
\verb|show|, \verb|EnrichmentProfiles-method| (\verb|show|),
show, PhenoProfiles-method (show), 34
show, Results-method (show), 34
show, SPADEResults-method (show), 34
SPADEResults (SPADEResults-class), 35
SPADEResults-class, 35
streamgraphViewer, 36
treeViewer, 36
unload.flowSet, 37
volcanoViewer, 37
```