Package 'SPADEVizR'

June 1, 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

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

LazyData true

Suggests knitr

VignetteBuilder knitr

RoxygenNote 5.0.1

R topics documented:

`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, show.on_device = TRUE)
```

Arguments

`AC`	an object of class AC (object returned by the 'computeAC()' function)
`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 shown (or just significant clusters)
`show.on_device`	a logical specifying if the respresentation will be displayed on device

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

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 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

Usage

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

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 character vector providing the sample names to used (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)
show.on_device	a logical specifying if the respresentation will be displayed on device

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, show.on_device = TRUE,
  verbose = FALSE)
```

Arguments

Results	a SPADEResults or Results object
conditions	conditions a named vector providing the correspondence between a sample name (in row names) and the condition of this sample
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 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
show.on_device	a logical specifying if the respresentation will be displayed on device
verbose	a logical specifying if the details of computation must be printed

Details

Cells clusters are colored based on theirs associated biological samples.

Value

a 'ggplot' object

---

buildCircles *Internal - Generate a circle representation*

Description

This function is used internally to generate a packed circles representation

Usage

```
buildCircles(circles, color = "grey80", class = NA, npoint = 100,
  limits = 30000, maxiter = 100)
```

Arguments

circles	a 2 column dataframe the clusters to be displayed and theirs sizes
color	a character specifying the color of the packed circles representation
class	a numeric specifyng the class number to be displayed
npoint	a numeric specifying the levels of details of polygones
limits	a numeric specifying the size of the coordinate system centered on (0,0)

Value

a ggplot2 object

---

buildCirclesLegend *Internal - Generate an legend for circles representation*

Description

This function is used internally to generate the legend of a packed circles representation

Usage

```
buildCirclesLegend(circles = data.frame(x = c(-29500, -19000, -8000, 3000,
  20000), y = c(20000, 20000, 20000, 20000, 20000), r = c(500, 1000, 2000, 5000,
  10000)), npoint = 100, limits = 30000)
```

Arguments

circles	a 3 colmuns data frame with the x, y coordinate of points and their raduis
npoint	a numeric specifying the levels of details of polygones
limits	a numeric specifying the size of the coordinate system centered on (0,0)

Value

a ggplot2 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 the number of cells 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.

`CCR-class` *Classification of clustering Results (CCR class) definition*

Description

The 'CR' 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 'classifyClusteringResults()' function.

Slots

`type` a character specifying if the classification is based on the phenotype profiles or on the enrichment profiles

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

`cluster.size` a numeric vector containing the number of cells for each cluster

`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 colunm and corresponding classe in the second colunm

---

`classificationViewer` *classificationViewer*

Description

Generate a graph representation of classified clusters

Usage

```
classificationViewer(CCR, show.on_device = TRUE)
```

Arguments

`CCR`	an object of class 'CCR' (object returned by the 'classifyclusteringResults()' function)
`show.on_device`	a logical specifying if the respresentation will be displayed on device

Details

Clusters of the same classe are shown using a circular graph. Circular graphs are sorted by the number of cluster in each class.

Value

a 'ggplot' object

`classifyClusteringResults` *Classification of clutering results based on the phenotype profiles or abundance profiles*

Description

Classifies clusters based on their phenotype profiles (expressions of markers) or abundance profiles (number of cells for each cluster).

Usage

```
classifyClusteringResults(Results, type = "phenotype",
  method = "hierarchical_h", method.parameter = NULL)
```

Arguments

`Results`	a Results or SPADEResults object
`type`	a character specifying if the classification is based on the phenotype profiles or on the abundance profiles
`method`	a character specifying the clustering method among one of those : "hierarchical_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 heigth 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 values 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.

Value

a S4 object of class 'CCR'

`computeClique` *Internal - Clique percolation classification*

Description

This function is used internally to classify clusters abundance 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 abundance 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 abundance 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

---

computeKmeans *Internal - Kmeans classification*

Description

This function is used internally to classify clusters abundance 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

---

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 *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

---

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, show.on_device = TRUE)
```

Arguments

CC	an object of class 'CC' (object returned by the 'computeCC()' function)
show.cluster.sizes	a logical specifying if dot sizes are proportional to cell counts
show.all_labels	a logical specifying if all cluster label must be show or just significant cluster
show.on_device	a logical specifying if the respresentation will be displayed on device

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.

Usage

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

Arguments

`Results`	a SPADEResults or Results object
`samples`	a character vector providing the sample names to used (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
`show.on_device`	a logical specifying if the respresentation will be displayed on device

Value

a 'ggplot' object

DAC-class *Differentially Abundant Clusters (DAC class) definition*

Description

The 'DAC' object is a S4 object containing the information related to the differentially abundant 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 'identifyDAC()' 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 the number of cells 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 DAC

`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 abundant.

distogramViewer *Visualization of marker co-expressions*

Description

Generate a distogram representation showing the marker co-expressions.

Usage

```
distogramViewer(Results, clusters = NULL, samples = NULL, markers = NULL,
  show.on_device = TRUE)
```

Arguments

Results	a SPADEResults or Results object
clusters	a character vector containing the clusters names to be use (by default all clusters will be used)
samples	a character vector providing the sample names to used (all samples by default)
markers	a character vector specifying the markers to be displayed
show.on_device	a logical specifying if the respresentation will be displayed on device

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

---

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.

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 'DAC'
export(object, filename = "export.txt")

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

## S4 method for signature 'CCR'
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 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
- "heatmap" (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. The 'clusters' parameter is required
- "pheno" (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
- "disto" (included by default):Display a distogram representation showing the marker co-expressions
- "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",
  "heatmap", "tree", "disto", "MDSclusters", "pheno"), clusters = NULL,
  markers = NULL, samples = NULL, assignments = NULL, conditions = NULL,
  stat.objects = list(), width = 30, height = 15, verbose = TRUE)
```

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)
samples	a character vector providing the sample names to used (all samples 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 plotable objects to be displayed in the report (object of class 'DAC', 'AC', 'CC' or 'CCR' accepted)
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 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 numeric matrix containing the markers expression categories
dendrogram.type	a character specifying the look of dendrograms ("rectangle" or "triangle", "rectangle" 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 is used internally to 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.

Usage

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

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 *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 *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

heatmapViewer *Visualization of all clusters phenotypes as an heatmap*

Description

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

Usage

```
heatmapViewer(Results, num = 5, show.on_device = TRUE)
```

Arguments

Results	a SPADEResults or Results object
num	a numeric value specifying the number of markers expression categories to use
show.on_device	a logical specifying if the respresentation will be displayed on device

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.

If the 'Results' parameter is a 'SPADEResults' object, markers used by SPADE to clustered cell populations are shown in bold.

Value

a list of 'ggplot' objects

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, samples, use.percentages = TRUE, method = "t.test",
  method.adjust = NULL, th.pvalue = 0.05, th.mean = 0)
```

Arguments

Results	a 'Results' or 'SPADEResults' object
samples	a character vector providing the sample names to used
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 clusters. 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
th.mean	a numeric specifying the abundance mean threshold

Value

a S4 object of class 'AC'

---

identifyCC *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 numerical phenotype
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
th.correlation	a numeric specifying the absolute value of the correlation coefficient threshold

Value

a S4 object of class 'CC'

identifyDAC *Identification of the Differentially Abundant Clusters*

Description

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

Usage

```
identifyDAC(Results, condition1, condition2, 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
condition1	a character vector providing the sample names defined as the first condition
condition2	a character vector providing the sample names defined as the 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
th.fc	a numeric specifying the fold-change threshold

Value

a S4 object of class 'DAC'

importResults *Import clustering results generated by other algorithms*

Description

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

Usage

```
importResults(cluster.abundances, cluster.phenotypes, th.min_cells = 50)
```

Arguments

`cluster.abundances`
: a dataframe of cells abundances with clusters in row and samples in column

`cluster.phenotypes`
: 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"

`th.min_cells`
: a numeric specifying the minimun number of cell in a cluster of a sample to take in account its phenotype

Details

The 'cluster.abundances' dataframe must be formated with the cluster names in rownames as following:

X	sample1	sample1
cluster1	749	5421
cluster2	450	412

The 'cluster.phenotypes' 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, th.min_cells = 50)
```

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)
th.min_cells	a numeric specifying the minimun number of cell in a cluster of a sample to take in account its phenotype

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.

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, show.on_device = TRUE, verbose = FALSE)
```

Arguments

Results	a SPADEResults or Results object
assignments	a 2 column data.frame with the sample 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
show.on_device	a logical specifying if the respresentation will be displayed on device
verbose	a logical specifying if the details of computation must be printed

Details

Time-points are sorted in a way that strings with embedded numbers are in the correct order

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.

Usage

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

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 abundances.

Usage

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

Arguments

Results	a SPADEResults or Results object
use.percentages	a logical specifying if the visualization should be performed on percentage
assignments	a 2 column data.frame with the samples names in row names providing firstly the biologicial condition and secondly the individuals 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
show.on_device	a logical specifying if the respresentation will be displayed on device

Details

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

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 'DAC'
names(x)

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

Arguments

x	a SPADEVizR object

Value

a character providing the name of the object

---

phenoViewer *Visualization of cluster phenotypes*

Description

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

Usage

```
phenoViewer(Results, clusters = NULL, samples = NULL, markers = NULL,
  show.mean = "both", show.on_device = TRUE, verbose = FALSE)
```

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	a character vector providing the sample names to used (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"
show.on_device	a logical specifying if the respresentation will be displayed on device
verbose	a logical specifying if the details of computation must be printed

Details

The ranges of value between marker bounds (using the 'bounds' slot) will be displayed using a grey ribbon.

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.

If the 'Results' parameter is a 'SPADEResults' object, markers used by SPADE to clustered cell populations are shown in bold.

Value

a list of 'ggplot' objects

---

plot *Graphical representation for some SPADEVizR objects*

Description

This function generates a graphical representation for 'AC', 'DAC', 'CC', 'CCR' and objects.

Usage

```
plot(x, y = NULL, ...)

## S4 method for signature 'DAC,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 'CCR,missing'
plot(x, y = NULL, ...)
```

Arguments

x	a 'AC', 'DAC', 'CC' and 'CCR' object
y	a supplementary parameter transmited respectively to 'abundantClustersViewer()', 'volcanoViewer()' or 'correlatedClustersViewer()' functions
...	some supplementaries parameters transmited respectively to abundantClustersViewer, volcanoViewer or correlatedClustersViewer functions

Value

a 'ggplot' object

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 'DAC'
print(x)

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

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

Arguments

x	a SPADEVizR object

Value

none

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 'bounds' dataframe stores extremum bounds (minimum and maximun) marker expressions for each marker

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

`bounds` a numeric data.frame containing the extremum bounds for each markers

show *Textual previews for SPADEVizR objects*

Description

Show a previews for a SPADEVizR 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 'DAC'
show(object)

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

## S4 method for signature 'CCR'
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.

The 'bounds' slot inihirited from 'Result' object is overriden by the 'SPADEResults' object. Indeed this slot contains in this case, the marker expression quantiles based on all cells in the place of extremun bounds.

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

`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. The 'clusters' parameter is required.

Usage

```
streamgraphViewer(Results, samples = NULL, clusters = NULL,
  use.relative = FALSE, show.on_device = TRUE)
```

Arguments

`Results`	a SPADEResults or Results object
`samples`	a character vector providing the sample names to used (all samples by default)
`clusters`	a character vector containing the clusters names to be vizualised
`use.relative`	a logical specifying if the visualization should be performed on relative abundance
`show.on_device`	a logical specifying if the respresentation will be displayed on device

Details

The order of samples in the 'samples' vector corespond to the order where the sample will be displayed

Value

a 'ggplot' object

`treeViewer` *Visualization of combined SPADE trees*

Description

Generates a tree representation showing combined SPADE trees.

Usage

```
treeViewer(SPADEResults, samples = NULL, highlight = NULL, marker = NULL,
  show.on_device = TRUE)
```

Arguments

`SPADEResults`	a SPADEResults object (Results object is not accepted)
`samples`	a character vector providing the sample names to used (all samples by default)
`highlight`	an AC, DAC or CC object to highligth identified significant clusters in the SPADE tree
`marker`	a character specifying the marker name to display
`show.on_device`	a logical specifying if the respresentation will be displayed on device

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 abundant clusters*

Description

Generates a Volcano plot representation showing for each cluster

Usage

```
volcanoViewer(DAC = NULL, fc.log2 = TRUE, show.cluster.sizes = TRUE,
  show.all_labels = FALSE, show.on_device = TRUE)
```

Arguments

`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
`show.on_device`	a logical specifying if the respresentation will be displayed on device
`DEC`	an object of class 'DAC' (object returned by the 'computeDAC()' function)

Details

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

Value

a 'ggplot' object

Index