Package 'CellVizR'

June 13, 2023

Encoding UTF-8

Version 1.0-0

Type Package

Title Visualization and statistical analyses of single-cell data using manifold representations

Description The profiling of biological samples at the single-cell level, using either high-dimensional cytometry or single-cell transcriptomics, is becoming more and more common. Such generated data are usually analyzed using manifold algorithms, such as UMAP, tSNE, or LargeVis, combined with cell clustering algorithms. Nevertheless, this is still challenging for non-bioinformatician experts to easily handle the whole pipeline of computational analyses with the purpose of answering specific biological questions. CellVizR is an R package that allows the visualization and statistical analyses of single-cell data using manifold algorithms and clustering methods. Especially, several key analysis steps are available to perform (i) data importation; (ii) manifold generation and visualization; (iii) cell cluster identification; (iv) characterization of cell clusters; (v) statistical analysis of cell cluster abundances; (vi) multivariate analysis using both unsupervised and supervised algorithms; (vii) quality controls of input files and generated results. CellVizR can import cell events from FCS, MTX or txt file formats using different transformation and down-sampling approaches. Manifold representations can be generated using the UMAP, tSNE or LargeVis algorithms to project cell events into a two-dimensionality space. The identification of cell clusters can be done using multiple clustering algorithms, depending on user's assumptions. The characteristics of cell clusters can be visualized using scatter plots, categorical heatmaps of marker expressions, or using parallel coordinate representations. Cell clusters having abundances different between biological conditions can be identified using several statistical tests. Statistical results can be visualized using volcano plots or heatmaps. Unsupervised and supersized analysis approaches can be conducted by users to appreciate the homogeneity/heterogeneity of biological conditions, and to identify cell population biomarkers in a multivariate manner. Additionally, CellVizR provides a workflow for asserting the quality of identified cell clusters using statistical approaches.

Author Gwendolyn MARGUERIT, Nicolas TCHITCHEK

Imports checkmate,
cluster,
concaveman,
cowplot,
dbscan,
dendextend,
diptest,

FactoMineR,
flowCore,
FNN,
ggdendro,
ggiraph,
ggnewscale,
ggplot2,
ggpubr,
ggrepel,
ggridges,
Gmedian,
gridExtra,
gtools,
kohonen,
MASS,
methods,
plyr,
RANN,
reshape,
reshape2,
rstatix,
Rtsne,
scales,
Seurat,
spade,
stats,
stringr,
uwot,
viridis

Suggests knitr

License GPL-3 — file LICENSE

VignetteBuilder knitr

biocViews Clustering, DataImport, FlowCytometry, Normalization, StatisticalMethod, Software, Visualization, SingleCell, Transcriptomics

RoxygenNote 7.2.1

R topics documented:

`assignMetadata` *Assigns meta-information to biological samples*

Description

This function aims to attach meta-information to biological samples.

For each biological sample, the biological individual, the biological condition and the time point can be specified for subsequent analyses.

Usage

```
assignMetadata(Celldata, metadata)
```

Arguments

`Celldata`	a Celldata object
`metadata`	a data.frame containing contextual information about the biological samples. This data.frame must have 3 columns specifying for each sample the associated individual (column named 'individual'), the biological condition (column named 'condition') and the time point (column named 'timepoint'). Rownames must correspond to biological samples imported within the Celldata object.

Value

a S4 object of class 'Celldata'

`Celldata-class` *Celldata class definition*

Description

The Celldata object is a S4 object containing all single-cell information.

Slots

`samples` a character vector containing the names of the biological samples

`raw.markers` a character vector containing the names of the raw markers

`matrix.expression.r` a data.frame containing the raw marker expressions of each cell

`matrix.expression` a data.frame containing the marker expressions of each cell

`manifold` a data.frame containing the manifold coordinates

`manifold.params` a list containing the parameters used for manifold generation

`identify.clusters` a vector containing the names of the identified cell clusters

`identify.clusters.params` a vector containing the parameters used for the identification of the cell clusters

`concave.hulls` a data.frame containing the coordinates of the concave hulls of each cluster

`matrix.cell.count` a data.frame containing the number of cells associated to each cluster for each sample

`matrix.abundance` a data.frame containing the percentage of cells associated to each cluster for each sample

`statistic` a data.frame containing the statistics of cell clusters

`metadata` a data.frame containing the metadata associated to each sample

computeStatistics *Computes differential analysis statistics for cell clusters*

Description

This function aims to identify of differentially abundant clusters.

Such clusters correspond to cell clusters having abundances statistically different between two biological conditions. The statistical test used for the comparisons can be defined by users. For each cluster, the p-value, log2 fold-change and effect size relative to the reference condition are computed. Statistical comparison can be performed in a paired and unpaired manner. Computed p-values can be corrected for multiple testing.

Usage

```
computeStatistics(
  Celldata,
  condition,
  ref.condition,
  comparison = "cmp",
  test.statistics = c("wilcox.test", "t.test"),
 p.adjust = c("none", "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr"),
  paired = FALSE
)
```

Arguments

Celldata	a Celldata object
condition	a character value providing the name of the condition to be compared
ref.condition	a character value providing the name of reference condition
comparison	a character value providing the name of comparison
test.statistics	a character value providing the type of statistical test to use. Possible values are: 'wilcoxon' or 't-test'
p.adjust	a character value providing the type of p-value adjustment method to use. Possible values are: 'none', 'holm', 'hochberg', 'hommel', 'bonferroni', 'BH', 'BY', 'fdr'
paired	a boolean value indicating if individuals are paired

Value

a S4 object of class 'Celldata'

createMetaclusters *Create metaclusters*

Description

This function aims to gathered multiple cell clusters to a large cell cluster

Usage

```
createMetaclusters(Celldata, clusters, metacluster.name)
```

Arguments

Celldata	a Celldata object
clusters	a character vector containing the identifiers of the clusters to gather
metacluster.name	a character value containing the name of the metacluster to create

Value

a Celldata object

deleteClusters *Delete cluster*

Description

This function aims to delete a set of cell clusters from this analysis

Usage

```
deleteClusters(Celldata, clusters)
```

Arguments

Celldata	a Celldata object
clusters	a character vector containing the identifiers of the clusters to delete

Value

a Celldata object

export *Exports cell expression profiles to TSV or FCS files*

Description

Exports cell expression profiles from a Celldata object to a tab-separated or FCS files.

Cell expression profiles can be exported for a set of given samples and for a set of given cell clusters

Usage

```
export(Celldata, filename, clusters = NULL, samples = NULL)

## S4 method for signature 'Celldata'
export(Celldata, filename, clusters = NULL, samples = NULL)
```

Arguments

`Celldata`	a Celldata object
`filename`	a character value providing the name of the output file
`clusters`	a character vector containing the identifiers of the cell clusters to export. By default, all clusters are extracted.
`samples`	a character vector containing the names of biological samples to export. By default, all samples are extracted.

Value

none

generateManifold *Generates a manifold of cell events*

Description

This function aims to generate a manifold representation for cell events stored in a Celldata object.

This function allows the use UMAP, t-SNE or LargeVis dimension reduction techniques. The whole set of cell markers or specific cell markers can be used during the dimensionality reduction process. The random seed can also be defined to allow the reproducibility of generated results.

Usage

```
generateManifold(
  Celldata,
  type = c("UMAP", "tSNE", "lvish"),
  markers = NULL,
  seed = 42,
  verbose = TRUE,
  ...
)
```

Arguments

Celldata	a Celldata object
type	a character value specifying the type of manifold to compute. Possible values are: 'UMAP' for Uniform Manifold Approximation and Projection, 'tSNE' for t-distributed Stochastic Neighbor Embedding, and 'lvish' for LargeVis
markers	a character vector providing the cell markers to use for the manifold generation
seed	a numeric value providing the random seed to use during stochastic operations
verbose	a boolean value indicating if computational details must be displayed on the console
...	other arguments passed on to manifold methods

Value

a S4 object of class 'Celldata'

---

identifyClusters *Identify cell cluster of having similar marker expressions*

Description

This function aims to identify cell clusters, which are groups of cells having similar expressions for selected markers, using different unsupervised clustering methods.

Several clustering methods are available such as kmeans, kmedian, clara, DBSCAN, HDBSCAN and SOM. The cell clustering can be performed on the manifold representation or based on marker expression.

Usage

```
identifyClusters(
  Celldata,
  space = c("manifold", "markers"),
  markers = NULL,
  method = c("kmeans", "kmedian", "clara", "DBSCAN", "SOM"),
  concavity = 2,
  length.threshold = 0,
  seed = 42,
  ...
)
```

Arguments

Celldata	a Celldata object
space	a character value containing the space of the clustering method to use. Possible values are: 'manifold' or 'markers'
markers	a character vector providing the cell markers to use for the manifold generation

`method`	a character value containing the name of the clustering method to use. Possible values are: 'kmeans', 'kmedian', 'clara', 'DBSCAN' and 'SOM'
`concavity`	a numeric value providing a relative measure of concavity for the computation of the concave hulls (please refer to the function 'concaveman' of the 'concaveman' package)
`length.threshold`	a numeric value providing a threshold of the segment length for the computation of the concave hulls (please refer to the function 'concaveman' of the concaveman package)
`seed`	a numeric value providing the random seed to use during stochastic operations
`...`	other arguments passed on to the methods

Details

For each identify cell cluster, the boundaries of cells belonging to this cluster are delineated using a concave hull

Value

a S4 object of class 'Celldata'

`import` *Imports of cell expression profiles from TSV or FCS files*

Description

This function aims to import acquired cell events from cytometric profiling into a Celldata object.

Input files can be tab-separated or FCS files. Different transformations can be applied such as logicle, arcsinh or logarithmic. Importantly, a downsampling of cell events can be performed using uniformly-based or density-based random selections. Cell marker having technical or biological biaises can be excluded during the import.

Usage

```
import(
  files,
  filetype = "fcs",
  transform = c("logicle", "arcsinh", "logarithmic", "none"),
  d.method = c("none", "uniform", "density"),
 parameters.method = list(exclude.pctile = 0.01, target.pctile = 0.05, target.number =
    NULL, target.percent = 0.1),
  exclude.markers = NULL,
  seed = 42
)
```

Arguments

files	a character vector specifying the path of the tab-separated or FCS files to load
filetype	a character vector specifying the format of the loaded files. By default, FCS is used
transform	a character value containing the type of the transformation to apply. Possible values are: 'logicle', 'arcsinh', 'logarithmic' or 'none'
d.method	a character value containing the type of the downsampling to apply. Possible values are: 'none', 'uniform' or 'density'
parameters.method	a list value containing the parameters to use for downsampling
exclude.markers	a character vector providing the marker names to be excluded during the import
seed	a numeric value providing the random seed to use during stochastic operations

Value

a S4 object of class 'Celldata'

---

importMTX *Imports of cell expression profiles from MTX files*

Description

This function aims to import acquired cell events from single-cell transcriptomic profiling into a Celldata object.

Usage

```
importMTX(count, cells, features, meta, sample.col)
```

Arguments

count	a character vector specifying the path of the count mtx file containing the count values
cells	a character vector specifying the path of the mtx file containing the cell ids
features	a character vector specifying the path of the mtx file containing the gene ids
meta	a character specifying the name of the tsv metadata file to load
sample.col	a character specifying the name of the colmun to use as biological sample in the metadata file

Value

a S4 object of class 'Celldata'

`performUpsampling` *Performs the upsampling of downsampled events*

Description

This function aims to perform the upsampling of downsampled events events based on an existing Celldata object and existing cell events stored in tab-separated or FCS files.

Importantly, the identification of cell clusters must have been performed prior to this operation.

Usage

```
performUpsampling(
  Celldata,
  files,
  transform = c("logicle", "arcsinh", "logarithmic", "none")
)
```

Arguments

`Celldata`	a Celldata object
`files`	a character vector providing the path of the tab-separated or FCS files
`transform`	a character value containing the type of the transformation to apply. Possible values are: 'logicle', 'arcsinh', 'logarithmic' or 'none'

Value

a S4 object of class 'Celldata'

`plot` *Plots graphics for all Celldata objects*

Description

Generates a graphical representation for a Celldata object. The displayed representation depends on the current analysis status of the Celldata object.

- If the manifold has not been calculated, then the number of cells per sample will be displayed

- If the manifold has been calculated but not the clustering, then the manifold representation will be displayed

- If the manifold and clustering have been calculated, then a heatmap of marker expressions will be displayed

Usage

```
## S4 method for signature 'Celldata,ANY'
plot(x)
```

Arguments

x	a Celldata object

Value

a ggplot2 object

`plotBoxplot` *Plots cell cluster abundances using a boxplot representation*

Description

This function aims to visualize and compare the cell cluster abundances for each biological condition using boxplot and jitter representations.

The abundance of a specific cell cluster or a set of cell clusters can be displayed. The representation can be restricted to a specific set of samples. Moreover, boxplot can be constructed based on sample meta information. Statistic can be computed for all comparisons.

Usage

```
plotBoxplot(
  Celldata,
  clusters,
  samples = NULL,
  observation = c("individual", "condition", "timepoint"),
  value.y = c("percentage", "absolute"),
  test.statistics = c("wilcox.test", "t.test"),
  paired = FALSE,
  hide.ns = TRUE
)
```

Arguments

Celldata	a Celldata object
clusters	a character vector containing the identifiers of the clusters to use
samples	a character vector containing the names of biological samples to use. By default, all samples are used
observation	a character value containing the parameters to use
value.y	a character value containing the parameters to use. Possible value are percentage or absolute.
test.statistics	a character value providing the type of statistical test to use. Possible values are: 'wilcox.test' or 't.test'
paired	a boolean value indicating if a paired or unpaired comparison should be applied
hide.ns	a boolean value indicating if non-significant p-value must be hidden

Value

a ggplot2 object

`plotCellCounts` *Plots the numbers of cells for each sample*

Description

This function aims to visualize the number of cells associated to each sample.

This representation displays the samples in the X-axis and the number of associated cells in the Y-axis. Several statistics can be computed and shown.

Usage

```
plotCellCounts(
  Celldata,
  stats = c("min", "median", "mean", "q75", "max"),
  samples = NULL,
  sort = TRUE
)
```

Arguments

`Celldata`	a Celldata object
`stats`	a character vector providing the statistics to display. Possible values are: 'min', 'median', 'mean', 'q75', 'max'
`samples`	a character vector containing the names of biological samples to use. By default, all samples are used
`sort`	a boolean value indicating if clusters must be sorted by the number associated sample

Details

The following statistic can be computed:

-'min' corresponds to the lowest number of cells within a data set

-'median' corresponds to the number of cells separates the lower half from the upper half within data set

-'mean' corresponds to the number of cells quantity shared within data set

-'q75' corresponds to the number of cells separates the quantiles 75

-'max' corresponds to the largest number of cells within a data set

Value

a ggplot2 object

plotClustersCounts *Plots the numbers of cells of each cluster*

Description

This function aims to visualize the number of cells associated to each cluster.

This representation displays the clusters in the X-axis and the total number of associated cells in the Y-axis.

Usage

```
plotClustersCounts(
  Celldata,
  clusters = NULL,
  sort = TRUE,
  legend.max.samples = 10
)
```

Arguments

Celldata	a Celldata object
clusters	a character vector containing the identifiers of the clusters to use. By default, all clusters are used
sort	a boolean value indicating if clusters must be sorted by the number associated cluster
legend.max.samples	a numerical value specifying the maximal number of samples to display in the graphical legend

Value

a ggplot2 object

plotCombineHM *Plots a combined expression and statistic heatmaps*

Description

This function aims to combine the expression and statistic heatmaps.

Usage

```
plotCombineHM(HM1, HM2)
```

Arguments

HM1	a ggplot object containing the expression heatmap
HM2	a ggplot object containing the statistic heatmap

Value

a ggplot2 object

`plotCompareClusters` *Plot compare cluster*

Description

This function aims to visualise the expression of a given marker in each cluster.

Usage

```
plotCompareClusters(Celldata, clusters1, clusters2)
```

Arguments

`Celldata`	a Celldata object
`clusters1`	a character vector containing the identifier of the cluster to use
`clusters2`	a character vector containing the identifier of the cluster to use

Value

a ggplot2 object

`plotCoordinatesClusters`
Plots of phenotype of cell clusters using parallels coordinates

Description

This function aims to visualize the characteristics of cell clusters using parallels coordinates. Each line in the representation corresponds to a biological sample for which marker/gene expressions are displayed.

Usage

```
plotCoordinatesClusters(
  Celldata,
  condition.samples = c("condition", "timepoint"),
  samples = NULL,
  clusters
)
```

Arguments

`Celldata`	a Celldata object
`condition.samples`	a character vector containing the variables to be studied for the samples. Possible values are: 'condition' or 'timepoint'
`samples`	a character vector containing the names of biological samples to use. By default, all samples are used
`clusters`	a character vector containing the identifiers of the clusters to use

Value

a ggplot2 object

`plotCoordinatesMarkers` *Plot a marker for each cluster using parallel coordinates*

Description

This function aims to visualise the expression of a given marker in each cluster using parallel coordinates. Each line in the representation corresponds to a biological sample for which marker/gene expressions are displayed.

Usage

```
plotCoordinatesMarkers(
  Celldata,
  condition.samples = c("condition", "timepoint"),
  samples = NULL,
  clusters = NULL,
  markers
)
```

Arguments

`Celldata`	a Celldata object
`condition.samples`	a character vector containing the variables to be studied for the samples. Possible values are: 'condition' or 'timepoint'
`samples`	a character vector containing the names of biological samples to use. By default, all samples are used
`clusters`	a character vector containing the identifiers of the clusters to use. By default, all clusters are used
`markers`	a character vector containing the name of the markers to use

Value

a ggplot2 object

plotDistogram *Plots of a distogram of marker co-expression*

Description

This function aims to visualize the pairwise co-expression between all markers using a distogram representation. Each tile corresponds to the co-expression between two markers and is gradient-colored based on the Pearson or Spearman correlation

Usage

```
plotDistogram(Celldata, clusters = NULL, method = c("pearson", "spearman"))
```

Arguments

Celldata	a Celldata object
clusters	a character vector containing the identifier of the cluster to use
method	a character value indicating the name of correlation method to use

Value

a ggplot2 object

plotHmAbundances *Plots an heatmap of cell cluster abundances*

Description

This function aims to visualize the abundances of cell clusters using an heatmap representation.

In such heatmap each column corresponds a cell cluster and the row corresponds the different samples. The heatmap can be restricted to specific cell clusters and samples. The levels of abundance of each sample in each cluster is represented using a color gradient scale. Abundance values can be centered and reduced.

Usage

```
plotHmAbundances(
  Celldata,
  clusters = NULL,
  samples = NULL,
  saturation = 2.5,
  rescale = FALSE
)
```

Arguments

Celldata	a Celldata object
clusters	a character vector containing the identifiers of the clusters to use. By default, all clusters are used
samples	a character vector containing the names of biological samples to use. By default, all samples are used
saturation	a numeric value providing the saturation threshold of cell cluster abundances
rescale	a boolean specifying if cell cluster abundances must be centered and reduced

Value

a ggplot2 object

---

plotHmExpressions *Plots an heatmap of cell marker expressions*

Description

This function aims to visualize the cell marker expressions for selected markers and clusters.

The mean of median marker expressions is computed for each cluster, and marker expressions displayed using a categorical heatmap (5 categories are defined by default). The range expression of each cell marker is discretized into several categories between bounds of marker expressions. Hierarchical clustering, represented by dendrogramm, can be computed on both marker and cluster levels.

Usage

```
plotHmExpressions(
  Celldata,
  markers = NULL,
  clusters = NULL,
  N.metaclusters = 1,
 method.hclust = c("ward.D", "ward.D2", "single", "complete", "average", "mcquitty",
    "median", "centroid"),
  nb.cat = 5,
  seed = 42
)
```

Arguments

Celldata	a Celldata object
markers	a character vector providing the marker names to use. By default, all markers are used
clusters	a character vector containing the identifiers of the clusters to use. By default, all clusters are used
N.metaclusters	a numeric value providing the number of metaclusters expected. By default, one metaclusters are used

`method.hclust`	a character value providing the agglomeration method to be used. Possible values are: 'ward.D', 'ward.D2', 'single', 'complete', 'average', 'mcquitty', 'median' or 'centroid' (please refer to the function 'hclust' of the 'stats' package)
`nb.cat`	a numeric specifying the number of categories to use
`seed`	a numeric value providing the random seed to use during stochastic operations

Value

a ggplot2 object

`plotHmStatistics` *Plots an heatmap of a statistical analysis results*

Description

This function aims to visualize the results of differential cell clusters analysis.

This representation displays statistical information for each cell cluster for a given comparison of samples. Different statistics can be visualized, such as the p-value, the log2(fold-change), and effect size.

Usage

```
plotHmStatistics(
  Celldata,
  clusters = NULL,
  statistics = c("pvalue", "lfc", "effsize"),
  saturation = 3
)
```

Arguments

`Celldata`	a Celldata object
`clusters`	a character vector containing the identifiers of the clusters to use. By default, all clusters are used
`statistics`	a character value providing the name of the statistic to display. Possible values are: 'pvalue' for p-value, 'lfc' for log2 fold change or 'eff' for effect size
`saturation`	a numeric value providing the saturation value for statistics to display

Value

a ggplot2 object

plotLDA *Plots a LDA representation based cell cluster abundances*

Description

This function aims to represent a Linear Discriminant Analysis representation based on cell cluster abundances

Usage

```
plotLDA(
  Celldata,
  levels = c("predictions", "coefficients"),
  ref.condition,
  condition,
  clusters = NULL
)
```

Arguments

Celldata	a Celldata object
levels	a character value containing the variable to be displayed. Possible values are: 'clusters' or 'samples'
ref.condition	a character value providing the name of reference condition
condition	a character value providing the name of the condition to be compared
clusters	a character vector containing the identifiers of the clusters to use. By default, all clusters are used

Value

xx

plotManifold *Plots a representation of a computed manifold*

Description

This function aims to visualize a computed manifold representation for given analysis.

This representation can be used on a Celldata object for which a manifold analysis has been performed.

If a cell clustering has been performed, then the clusters are delineated using concave hulls. Additionally, the manifold can be colored based on the local cell density or marker expressions. It is possible to centre and reduce the values of expressions.

Usage

```
plotManifold(
  Celldata,
  markers = "density",
  samples = NULL,
  scale = FALSE,
  quant.low = 0.05,
  quant.high = 0.95
)
```

Arguments

Celldata	a Celldata object
markers	a character value providing the name of the marker to use for the colouring. By default, cells are colored based on their local density
samples	a character vector containing the names of biological samples to use
scale	a boolean value specifying if expression values must be rescaled
quant.low	a numeric value providing the number of first quantile
quant.high	a numeric value providing the number of last quantile

Value

a ggplot2 object

---

plotMarkerDensity *Plots of phenotype of identified cell clusters*

Description

This function aims to visualize the density of marker/gene expressions for a set of given clusters.

Usage

```
plotMarkerDensity(
  Celldata,
  clusters,
  quant.low = 0.05,
  quant.high = 0.95,
  dip.th = 0.01
)
```

Arguments

Celldata	a Celldata object
clusters	a character vector containing the identifier of the cluster to use
quant.low	a numeric value providing the value of first quantile in the color gradiant
quant.high	a numeric value providing the value of last quantile in the color gradiant
dip.th	a numeric value specifying the p-value threshold of the Hartigan's dip test to consider non-unimodal clusters

Value

a ggplot2 object

`plotMDS` *Plots a MDS representation based on cell cluster abundances*

Description

This function aims to visualize the similarities between samples or clusters based on their abundances, using a Multidimensional Scaling representation. Each dot represents a sample or a cluster and the distances between the dots are proportional to the Euclidean distance between these objects. The representation can be restricted to specific cell clusters and samples. In addition, it is possible to choose the levels displayed, clusters or samples.

Usage

```
plotMDS(
  Celldata,
  matrix = c("abundance", "expression"),
  levels = c("clusters", "samples"),
  condition.samples = c("condition", "timepoint"),
  clusters = NULL,
  samples = NULL,
  plot.text = TRUE
)
```

Arguments

`Celldata`	a Celldata object
`matrix`	a character vector containing the matrix to be studied. Possible values are: 'abundance' or 'expression'
`levels`	a character value containing the variable to be displayed. Possible values are: 'clusters' or 'samples'
`condition.samples`	a character vector containing the variable to be studied for the samples. Possible values are: 'condition' or 'timepoint"
`clusters`	a character vector containing the identifiers of the clusters to use. By default, all clusters are used
`samples`	a character vector containing the names of biological samples to use. By default, all samples are used
`plot.text`	a boolean value specifying if adds text directly at the plot

Value

a ggplot2 object

`plotPCA` *Plots a PCA representation based cell cluster abundances*

Description

This function aims to represent a Principal Component Analysis representation based on cell cluster abundances. In such representation, clusters or samples are positioned based on computed principal components. The representation can be displayed based on specific principal components. The representation can be restricted to specific cell clusters and samples. In addition, it is possible to choose the levels displayed, clusters or samples.

Usage

```
plotPCA(
  Celldata,
  levels = c("both", "clusters", "samples"),
  clusters = NULL,
  samples = NULL,
  components = c(1, 2),
  condition.samples = c("condition", "timepoint"),
  cor.radius.th = 0.6,
  plot.text = TRUE
)
```

Arguments

`Celldata`	a Celldata object
`levels`	a character value containing the variable to be displayed. Possible values are: 'both', 'clusters' or 'samples'
`clusters`	a character vector containing the identifier of the cluster to use. By default, all clusters are used
`samples`	a character vector containing the names of biological samples to use. By default, all samples are used
`components`	a numeric vector providing the components to display
`condition.samples`	a character vector containing the variable to be studied for the samples. Possible values are: 'condition' or 'timepoint"
`cor.radius.th`	a numeric value specifying the radius of the correlation plot radius
`plot.text`	a boolean value specifying if adds text directly at the plot

Value

a ggplot2 object

plotScatter *Plots of a scatter plot of marker co-expression*

Description

This function aims to visualize co-expression between two markers using a scatter representation

Usage

```
plotScatter(Celldata, marker1, marker2, samples = NULL, clusters = NULL)
```

Arguments

Celldata	a Celldata object
marker1	a character value specifying the first marker to be visualized
marker2	a character value specifying the second marker to be visualized
samples	a character vector containing the names of biological samples to use. By default, all samples are used
clusters	a character vector containing the identifiers of the clusters to use. By default, all clusters are used

Value

a ggplot2 object

plotVolcano *Plots of a volcano plot of statistical analysis*

Description

This function aims to visualize the results of a differentially abundant a analysis using a Volcano plot.

In such representation, each in dot corresponds to a cell cluster and dots are positioned in two dimensional space where the X-axis represents the log2(fold-change) and the Y-axis represents the -log10 of the p-value. Un horizontal line is displayed accordingly to the p-value threshold and to vertical lines are displayed accordingly to the fold-change threshold.

Usage

```
plotVolcano(Celldata, comparison, th.pv = 1.3, th.fc = 1.5, plot.text = TRUE)
```

Arguments

Celldata	a Celldata object
comparison	a character value containing the comparison to study
th.pv	a numeric value containing the p-value threshold to use
th.fc	a numeric value containing the fold-change threshold to use
plot.text	a boolean value specifying if adds text directly at the plot

Value

a ggplot2 object

`print` *Prints information for a given Celldata object*

Description

Prints a preview for a Celldata object.

Usage

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

Arguments

`x`	a Celldata object

Value

none

`QCCorrelationManifold` *QC for dimensionality reduction : Computes correlation between pairwise distances in the high-dimensional space and in the embedding*

Description

CPD quantifies preservation of the global, or macroscropic structure. A performant RD should give a boxplot with linear relationships betweens pairwise distance in the different data spaces

Usage

```
QCCorrelationManifold(
  Celldata,
  downsampling = 1000,
  method = c("pearson", "kendall", "spearman"),
  plot.device = TRUE
)
```

Arguments

`Celldata`	a S4 object of class 'Celldata'
`downsampling`	a numeric being the sample size used for computation
`method`	a character being the method use to compute correlation. Possible values are : "pearson", "kendall", "spearman". Default value is "pearson"
`plot.device`	a boolean value specifying if result representation must be displayed

Value

a list containing correlation coefficient and boxplot of pairwise distances

---

`QCKncManifold` *QC for dimensionality reduction : Computes proportion of nearest clusters preservation in reduced dimension in comparison to high dimension*

Description

The fraction of k-nearest class (clusters) means in the original data that are preserved as k-nearest class means in the embedding. it computes the class means only and averages across all classes. KNC quantifies preservation of the mesoscopic structure.

Usage

```
QCKncManifold(Celldata, KNC = 5, downsampling = 50)
```

Arguments

`Celldata`	a S4 object of class 'Celldata'
`KNC`	a numeric being the number of nearest classes to compute
`downsampling`	a numeric being the sample size used for computation

Value

a numeric being proportion

`QCKnnManifold` *QC for dimensionality reduction : Computes proportion of nearest neighbours preservation in reduced dimension in comparison to high dimension*

Description

The fraction of k-nearest neighbours in the original high-dimensional data that are preserved as k-nearest neighbours in the embedding It compute the average across all n points. KNN quantifies preservation of the local, or microscopic structure.

Usage

```
QCKnnManifold(Celldata, KNN = 5, downsampling = 50)
```

Arguments

`Celldata`	a S4 object of class 'Celldata'
`KNN`	a numeric being the number of nearest neighbors to compute
`downsampling`	a numeric being the sample size used for computation

Value

a numeric being proportion

`QCMarkerNames` *Verifies the consistency of the marker names within cell event files*

Description

This function aims to check the consistency of marker names across multiple tab-separated or FCS files.

Additionally, the number of cells associated to each sample is displayed.

Usage

```
QCMarkerNames(files)
```

Arguments

`files`	a character vector specifying the path of the tab-separated or FCS files to check

Value

a data.frame containing the marker names and the associated number of cells for each sample (rownames = samples and colnames = markers)

`QCMarkerRanges` *Verifies the consistency of marker expressions integrity within cell event files*

Description

This function aims to check the consistency of marker expressions ranges across multiple tab-separated or FCS files.

The marker expressions ranges are calculated based on the user-defined quantiles.

Usage

```
QCMarkerRanges(files, probs = c(0.05, 0.95))
```

Arguments

`files`	a character vector specifying the path of the FCS files to verified
`probs`	a numerical vector providing the quantiles used to define marker expressions ranges

Value

a list containing two data.frame for the lower and upper marker expression ranges (rownames = samples and colnames = markers)

QCSmallClusters *Computes the percentage of cell clusters with low number of cells*

Description

This function aims to compute and show cell clusters having a number of associated cells lower than a specific threshold.

Usage

```
QCSmallClusters(Celldata, th.size = 50, plot.device = TRUE)
```

Arguments

Celldata	a Celldata object
th.size	a numeric value providing the minimum number of cells needed for a cluster to be considered a small cluster
plot.device	a boolean value specifying a results representation must be displayed

Value

a numerical value corresponding to the percentage of cell cluster with low number of cells

QCUniformClusters *Computes the percentage of clusters with uniform phenotypes*

Description

This function aims to identify and show cell clusters having a uniform phenotype.

A uniform cluster corresponds to a cluster that have a unimodal expression and a low spread of expression for all its markers.

Usage

```
QCUniformClusters(
  Celldata,
  uniform.test = c("both", "uniform", "IQR"),
  th.pvalue = 0.05,
  th.IQR = 2,
  plot.device = TRUE
)
```

Arguments

`Celldata`	a Celldata object
`uniform.test`	a character providing the name of test assessment to perform. Possible value are : 'both', 'uniform', 'IQR'
`th.pvalue`	a numeric value providing the p-value threshold of the Hartigan's dip test (unimodal if pvalue ¿ th.pvalue)
`th.IQR`	a numeric value providing the IQR (interquartile range) threshold to assume a distribution as uniform
`plot.device`	a boolean value specifying if result representation must be displayed

Details

-'uniform' corresponds to the verification of the unimodal distribution of markers with a Hartigans test

-'IQR' corresponds to the verification of the distribution of markers so that they are not below the IQR threshold (interquantile range)

-'both' corresponds to the combination of the two parameters : uniform and IQR

Value

a numerical value corresponding to the percentage of cell cluster with unimodal expression and a low spread

---

`renameMarkers` *Renames markers within a Celldata object*

Description

This function aims to rename cell markers stored within a Celldata object.

This function is interesting to remove the names of the fluorochromes or metals recorded during the acquisition process.

Usage

```
renameMarkers(Celldata, marker.names)
```

Arguments

`Celldata`	a Celldata object
`marker.names`	a character vector providing the new marker names to use

Value

a S4 object of class 'Celldata'

selectSamples *Select samples based on metadata information*

Description

This function aims to select biological samples of interest based on provided metadata.

Usage

```
selectSamples(Celldata, individual = NULL, condition = NULL, timepoint = NULL)
```

Arguments

Celldata	a Celldata object
individual	a character vector indicating the individual to select
condition	a character vector indicating the biological condition to select
timepoint	a character vector indicating the timepoint to select

Value

a character vector

show *Prints information for a Celldata objects*

Description

Shows a preview for a Celldata object

Usage

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

Arguments

object	a Celldata object

Value

none

Index