Package 'UMAPVizR'

May 16, 2022

Encoding UTF-8

Version 1.0-1

Type Package

Title Visualization and statistical analysis of high dimensional cytometry data using manifold representations

Description

Cytometry data are now classically analyzed using non linear dimensionality reduction approaches, but it is still challenging to easily handle the whole pipeline of computational analyses.
UMAPVizR allows the statistical analysis and visualization of high dimensional cytometry data using manifold algorithms and clustering methods. Especially, several key analysis steps are available to perform data importation, manifold generation, cell cluster identification, statistical analyses, cluster visualization, and quality controls of generated results.
UMAPVizR can import cell events from FCS or txt file formats using different transformation, down-sampling, and normalization approaches.
Manifold representations can be generated using the UMAP, tSNE or LargeVis algorithms to project cell events into a lower 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 plot, categorical heatmap of marker expressions, or using parallel coordinates representations. Cell clusters having abundances differently expressed 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, UMAPVizR provides a workflow for asserting the quality of identified cell clusters using statistical approaches.

Author Gwendolyn MARGUERIT, Nicolas TCHITCHEK

Imports checkmate,
cluster,
concaveman,
dbscan,
dendextend,
diptest,
FactoMineR,
flowCore,
FNN,
fpc,

ggdendro,
Gmedian,
ggnewscale,
ggplot2,
ggpubr,
ggrepel,
ggridges,
gridExtra,
gtools,
kohonen,
MASS,
methods,
plyr,
reshape,
reshape2,
rstatix,
Rtsne,
scales,
stats,
stringr,
uwot

Suggests knitr

License GPL-3 | file LICENSE

VignetteBuilder knitr

biocViews FlowCytometry, Visualization, StatisticalMethod, Clustering, Software

RoxygenNote 7.1.2

R topics documented:

abs.proj . . . 3
assignMetadata . . . 4
computeCellCounts . . . 4
computeCellDensities . . . 5
computeClusterAbundances . . . 5
computeConcaveHulls . . . 6
computeMarkerMedians . . . 6
computeSmallClusters . . . 7
computeStatistics . . . 7
computeUniformClusters . . . 8
createFlowframe . . . 9
export . . . 9
generateManifold . . . 10
generateManifoldlvish . . . 10
generateManifoldtSNE . . . 11
generateManifoldUMAP . . . 12
identifyClusters . . . 12
import . . . 13
performUpsampling . . . 14
plot . . . 15
plotBoxplot . . . 15

plotCellCounts . . . 16
plotClustersCounts . . . 17
plotCombineHM . . . 18
plotCoordinates . . . 18
plotDistogram . . . 19
plotHmAbundances . . . 19
plotHmExpressions . . . 20
plotHmStatistics . . . 21
plotLDA . . . 21
plotManifold . . . 22
plotMDS . . . 22
plotPCA . . . 23
plotPhenoClusters . . . 24
plotScatter . . . 25
plotSmallClusters . . . 25
plotUniformClusters . . . 26
plotVolcanoPlot . . . 26
print . . . 27
QCMarkerNames . . . 27
QCMarkerRanges . . . 28
QCSmallClusters . . . 28
QCUniformClusters . . . 29
renameMarkers . . . 30
rescaleMarkerExpressions . . . 30
show . . . 31
UMAPdata-class . . . 31

Index **32**

`abs.proj` *Internal - Rescales numeric matrix for projection*

Description

This functions is used internally xxx

Usage

```
## S3 method for class 'proj'
abs(proj, abs, quant.low, quant.high)
```

Arguments

`proj`	a data.frame providing the manifold representation
`abs`	a boolean value providing xx
`quant.low`	a numeric value providing the number first quantile
`quant.high`	a numeric value providing the number last quantile

Value

a numeric vector containing scale limits

assignMetadata *Assigns meta-information about biological samples*

Description

This function aims to attach meta-information to each biological sample.

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

Usage

```
assignMetadata(UMAPdata, metadata)
```

Arguments

UMAPdata	a UMAPdata 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')

Value

a S4 object of class 'UMAPdata'

computeCellCounts *Internal - Computes the number of cells for each cluster*

Description

This function is used internally to computes the number of cells for each cluster.

Usage

```
computeCellCounts(proj, clusters, samples)
```

Arguments

proj	a data.frame providing the manifold representation with two columns
clusters	a character vector providing the cluster to analyse the associated with cell cluster
samples	a character vector providing for each cell the associated biological sample

Value

a data.frame containing the numbers of cells associated for each cluster for each sample (rownames = clusters / colnames = samples)

computeCellDensities *Internal - Computes the cell density within the manifold*

Description

This function is used internally to compute the cell density

Usage

```
computeCellDensities(proj, n)
```

Arguments

proj	a data.frame providing the manifold representation with two columns
n	a numerical providing the number a grid points in each direction (please refer to the function 'kde2d' of the 'MASS' package)

Value

a numeric vector containing the computed cell density

computeClusterAbundances
Internal - Computes the abundances for each cell cluster

Description

This function is used internally to computes the abundance of each cluster for each sample.

Usage

```
computeClusterAbundances(count)
```

Arguments

count	a data.frame providing the numbers of cells associated to each cluster for each sample

Value

a data.frame containing the abundance of cells to each clusters for each sample

computeConcaveHulls *Internal - Computes the concave hulls for identified cell clusters*

Description

This function is used internally to computes the concave hulls each cell cluster. Each concave hulls correspond the boundaries of the cells cluster with manifold representation.

Usage

```
computeConcaveHulls(proj, clusters, concavity = 2, length.threshold = 0)
```

Arguments

proj	a data.frame providing the manifold representation
clusters	a character vector providing id of cell clusters for which the concave hulls must be computed
concavity	a numeric value providing a relative measure of concavity (please refer to the function 'concaveman' of the 'concaveman' package)
length.threshold	a numeric value providing a threshold for the segment length (please refer to the function 'concaveman' of the concaveman package)

Value

a data.frame containing the concave hulls for each cluster (dim1, dim2, clusters)

computeMarkerMedians *Internal - Computes marker expression medians for each cluster*

Description

This function is used internally to computes marker expression medians for each cluster.

Usage

```
computeMarkerMedians(exprs)
```

Arguments

exprs	a data.frame containing the marker expressions for each cell

Value

a data.frame containing the median of marker expressions for each cluster (clusters, markers, medians)

`computeSmallClusters` *Internal - Computes the percentage of clusters with low number of cells*

Description

This function is used internally to compute the percentage of clusters having a number of associated cells lower than a specific threshold.

Usage

```
computeSmallClusters(UMAPdata, th.size)
```

Arguments

`UMAPdata`	a UMAPdata object
`th.size`	a numeric value providing the minimum number of cells needed for a cluster to be considered a small cluster

Value

a list containing QC information for small clusters. Returns a data.frame with a boolean value indicating if the number of associated cells is greater or less than the threshold. In addition, returns the percentage of the calculation.

`computeStatistics` *Computes differential analysis statistics for cell clusters*

Description

This function aims to compute the statistics of Differentially Abundant Clusters 'DAC'.

DAC 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.

Usage

```
computeStatistics(
  UMAPdata,
  condition,
  ref.condition,
  test.statistics = c("wilcoxon", "t-test"),
  paired = c("paired", "unpaired")
)
```

Arguments

UMAPdata	a UMAPdata 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
test.statistics	a character value providing the type of statistical test to use. Possible values are: 'wilcoxon' or 't-test'
paired	a boolean value indicating if a paired or unpaired comparison should be applied

Value

a S4 object of class 'UMAPdata'

---

computeUniformClusters

Internal - Computes percentage of clusters with uniform phenotype

---

Description

This function is used internally to identify cell clusters that have a non-uniform phenotype. A uniform cluster corresponds to clusters that have a unimodal expression and having a low spread of expression for all the markers to compose it.

Usage

```
computeUniformClusters(UMAPdata, uniform.test, th.pvalue, th.IQR)
```

Arguments

UMAPdata	a UMAPdata 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

Value

a list containing QC information for small clusters. Returns a data.frame with a boolean value indicating xxx. In addition, returns the percentage of the calculation.

createFlowframe *Internal - Creates of a FlowFrame object*

Description

This function is used internally to create a flowframe object with the purpose extracting marker intensities to FCS files.

Usage

```
createFlowframe(intensities)
```

Arguments

intensities	a data.frame providing the cell profile intensities

Value

a flowframe object containing the marker expression

---

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

Description

Exports cell expresion profiles from a UMAPdata 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(UMAPdata, filename, clusters = NULL, samples = NULL)

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

Arguments

UMAPdata	a UMAPdata 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 UMAPdata object.

This function allows the use of several non-linear dimension reduction techniques such as UMAP, t-SNE or LargeVis. The whole set of cell markers or specific cell markers can be used during the dimensionality reduction process.

Usage

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

Arguments

UMAPdata	a UMAPdata 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 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 methods

Value

a S4 object of class 'UMAPdata'

generateManifoldlvish *Internal - Generates a LargeVis manifold of cell events*

Description

This function is used internally to compute a manifold representation based on the LargeVis algorithms

Usage

```
generateManifoldlvish(exprs, seed, verbose, ...)
```

Arguments

exprs	a data.frame containing the marker expressions
seed	a numeric value providing the random seed to use in stochastic operation
verbose	a boolean value indicating if computational details must be displayed on the console
...	Other arguments passed on to methods

Value

a data.frame containing the manifold coordinates

---

generateManifoldtSNE *Internal - Generates a t-SNE manifold of cell events*

Description

This function is used internally to compute a manifold representation based on the t-SNE algorithms.

Usage

```
generateManifoldtSNE(exprs, seed, verbose, ...)
```

Arguments

exprs	a data.frame containing the marker expressions
seed	a numeric value providing the random seed to use in stochastic operation
verbose	a boolean value indicating if computational details must be displayed on the console (please refer to the function 'Rtsne' of the 'Rtsne' package)
...	Other arguments passed on to methods

Value

a data.frame containing the manifold coordinates

generateManifoldUMAP *Internal - Generates a UMAP manifold of cell events*

Description

This function is used internally a manifold representation based on the UMAP algorithms.

Usage

```
generateManifoldUMAP(exprs, seed, verbose, ...)
```

Arguments

exprs	a data.frame providing the marker expressions
seed	a numeric value providing the random seed to use in stochastic operation
verbose	a boolean value indicating if computational details must be displayed on the console
...	Other arguments passed on to methods

Value

a data.frame containing the manifold coordinates

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

Arguments

UMAPdata	a UMAPdata object
space	a character value containing the space of clustering method to use. Possible values are: 'manifold' or 'markers'
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 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 'UMAPdata'

---

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

Description

This function aims to import acquired cell events into a UMAPdata 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"),
  downsampling = NULL,
  d.method = c("uniform", "density"),
  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 xxx
transform	a character value containing the type of the transformation to apply. Possible values are: 'logicle', 'arcsinh', 'logarithmic' or 'none'
downsampling	a numeric value providing the number of cells to downsample for each sample
d.method	a character value containing the type of the dowsampling to apply. Possible values are: 'uniform' or 'density'
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 'UMAPdata'

---

performUpsampling *Performs the upsampling of downsampled events*

Description

This function aims to perform upsample downsampled events based on an existing UMAPdata 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(UMAPdata, files)
```

Arguments

UMAPdata	a UMAPdata object
files	a character vector providing the path of the tab-separated or FCS files

Value

a S4 object of class 'UMAPdata'

plot *Plots graphics for all UMAPdata objects*

Description

Generates a graphical representation for a UMAPdata object. The displayed representation depends on the current analysis status of the UMAPdata 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 'UMAPdata,ANY'
plot(x)
```

Arguments

x	a UMAPdata 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 resticted 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(
  UMAPdata,
  clusters,
  samples = NULL,
  observation = c("individual", "condition", "timepoint"),
  test.statistics = c("wilcox.test", "t.test"),
  paired = FALSE,
  hide.ns = TRUE
)
```

Arguments

UMAPdata	a UMAPdata 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
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 xxx

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

Arguments

UMAPdata	a UMAPdata 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 for each clusters*

Description

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

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

Usage

```
plotClustersCounts(UMAPdata, clusters = NULL, sort = TRUE)
```

Arguments

`UMAPdata`	a UMAPdata 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

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

plotCoordinates *Plots parallel coordinates*

Description

This function aims to visualize xxx

Usage

```
plotCoordinates(UMAPdata)
```

Arguments

UMAPdata	a UMAPdata object

Value

xx

`plotDistogram` *Plots of marker co-expression*

Description

This function aims to visualize xxx

Usage

```
plotDistogram(UMAPdata, clusters = NULL, show.on.device = TRUE)
```

Arguments

`UMAPdata`	a UMAPdata object
`clusters`	a character vector containing the identifier of the cluster to use
`show.on.device`	a numeric value containg xxx

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 he 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(
  UMAPdata,
  clusters = NULL,
  samples = NULL,
  saturation = 2.5,
  rescale = FALSE
)
```

Arguments

`UMAPdata`	a UMAPdata 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. To hierarchical clustering, shown using dendrogramm, can be computed on both marker and cluster levels.

Usage

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

Arguments

`UMAPdata`	a UMAPdata 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
`method.hclust`	a character value providing the agglomeration method to be use. 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(
  UMAPdata,
  clusters = NULL,
  statistics = c("pvalue", "lfc", "statistic"),
  saturation = 3
)
```

Arguments

UMAPdata	a UMAPdata 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(UMAPdata)
```

Arguments

UMAPdata	a UMAPdata object

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 UMAPdata object for which a manifold analysis has been performed.

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

Usage

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

Arguments

UMAPdata	a UMAPdata 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 specifing if expression calue 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

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(
  UMAPdata,
  levels = c("clusters", "samples"),
  clusters = NULL,
  samples = NULL
)
```

Arguments

UMAPdata	a UMAPdata object
levels	a character value containing the variable to be displayed. Possible values are: 'clusters' or 'samples'
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

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(
  UMAPdata,
  levels = c("clusters", "samples", "both"),
  clusters = NULL,
  samples = NULL,
  components = c(1, 2),
  cor.radius.th = 0.6,
  cluster.label.size = 3,
  sample.label.size = 3,
  cluster.dot.size = 2,
  sample.dot.size = 3
)
```

Arguments

UMAPdata	a UMAPdata object
levels	a character value containing the variable to be displayed. Possible values are: '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
cor.radius.th	a numeric value containing
cluster.label.size	a numeric value containing xxx
sample.label.size	a numeric value containing xxx
cluster.dot.size	a numeric value containing xxx
sample.dot.size	a numeric value containing xxx

Value

a ggplot2 object

---

plotPhenoClusters *Plots of phenotype of identified cell clusters*

Description

This function aims to visualize xxx

Usage

```
plotPhenoClusters(
  UMAPdata,
  cluster,
  quant.low = 0.05,
  quant.high = 0.95,
  dip.th = 0.01
)
```

Arguments

UMAPdata	a UMAPdata object
cluster	a character vector containing the identifier of the cluster to use
quant.low	a numeric value containg xxx
quant.high	a numeric value containing xxx
dip.th	a numeric value containing xxx

Value

a ggplot2 object

`plotScatter` *Plots of scatter*

Description

This function aims to visualize xxx

Usage

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

Arguments

`UMAPdata`	a UMAPdata object
`marker1`	a character value containing xxx
`marker2`	a character value containing xxx
`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

xx

`plotSmallClusters` *Internal - Plots a representation of QC for small clusters*

Description

This function is used internally to create a representation showing the fraction of clusters having a number associated cell lower than a specific threshold.

Usage

```
plotSmallClusters(values.small)
```

Arguments

`values.small`	a list providing the small cluster QC information. Such as data.frame containing the boolean values and the percentage computed.

Value

a ggplot2 object

plotUniformClusters *Internal - Plots a representation of QC for uniform phenotype cluster*

Description

This function is used internally to create a graphic representation.

Usage

```
plotUniformClusters(values_uniform)
```

Arguments

values_uniform	a list providing the uniform cluster QC information. Such as data.frame containing the boolean values and the percentage computed.

Value

a ggplot2 object

plotVolcanoPlot *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

```
plotVolcanoPlot(UMAPdata, th.pv = 1.3, th.fc = 1.5)
```

Arguments

UMAPdata	a UMAPdata object
th.pv	a numeric value containing the p-value threshold to use
th.fc	a numeric value containing the fold-change threshold to use

Value

a ggplot2 object

print *Prints information for a given UMAPdata object*

Description

Prints a preview for a UMAPdata object.

Usage

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

Arguments

x	a UMAPdata object

Value

none

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

Arguments

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

Arguments

`UMAPdata`	a UMAPdata 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 UMAPdata object*

Description

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

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

Usage

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

Arguments

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

Value

a S4 object of class 'UMAPdata'

`rescaleMarkerExpressions` *Internal - Rescales marker expression by quantile*

Description

This function is used internally to rescale the marker expression by the quantile method.

Usage

```
rescaleMarkerExpressions(exprs, quant.low = 0.05, quant.high = 0.95)
```

Arguments

`exprs`	a data.frame containing the marker expressions for each cluster
`quant.low`	a numeric value providing the value of the first quantile
`quant.high`	a numeric value providing the value of the last quantile

Value

a data.frame containing quantile rescale marker expressions

show *Prints information for a UMAPdata objects*

Description

Shows a preview for a UMAPdata object.

Usage

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

Arguments

object	a UMAPdata object

Value

none

UMAPdata-class *UMAPdata class definition*

Description

The UMAPdata object is a S4 object containing all cytometry expressions.

Slots

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

`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 creation

`recognize.clusters` a vector containing the identifiers of cell clusters

`recognize.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 cell cluster of the concave hulls for 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

Index

abs.proj, 3
assignMetadata, 4

computeCellCounts, 4
computeCellDensities, 5
computeClusterAbundances, 5
computeConcaveHulls, 6
computeMarkerMedians, 6
computeSmallClusters, 7
computeStatistics, 7
computeUniformClusters, 8
createFlowframe, 9

export, 9
export,UMAPdata-method *(*export*)*, 9

generateManifold, 10
generateManifoldlvish, 10
generateManifoldtSNE, 11
generateManifoldUMAP, 12

identifyClusters, 12
import, 13

performUpsampling, 14
plot, 15
plot,UMAPdata,ANY-method *(*plot*)*, 15
plotBoxplot, 15
plotCellCounts, 16
plotClustersCounts, 17
plotCombineHM, 18
plotCoordinates, 18
plotDistogram, 19
plotHmAbundances, 19
plotHmExpressions, 20
plotHmStatistics, 21
plotLDA, 21
plotManifold, 22
plotMDS, 22
plotPCA, 23
plotPhenoClusters, 24
plotScatter, 25
plotSmallClusters, 25
plotUniformClusters, 26
plotVolcanoPlot, 26

print, 27
print,UMAPdata-method *(*print*)*, 27

QCMarkerNames, 27
QCMarkerRanges, 28
QCSmallClusters, 28
QCUniformClusters, 29

renameMarkers, 30
rescaleMarkerExpressions, 30

show, 31
show,UMAPdata-method *(*show*)*, 31

UMAPdata *(*UMAPdata-class*)*, 31
UMAPdata-class, 31