Package 'conclus'

April 10, 2022

Title ScRNA-seq Workflow CONCLUS - From CONsensus CLUSters To A Meaningful CONCLUSion

Version 1.2.4

Description CONCLUS is a tool for robust clustering and positive marker features selection of single-cell RNA-seq (sc-RNA-seq) datasets. It takes advantage of a consensus clustering approach that greatly simplify sc-RNA-seq data analysis for the user. Of note, CONCLUS does not cover the preprocessing steps of sequencing files obtained following next-generation sequencing. CONCLUS is organized into the following steps:
Generation of multiple t-SNE plots with a range of parameters including different selection of genes extracted from PCA. Use the Density-based spatial clustering of applications with noise (DBSCAN) algorithm for idenfication of clusters in each generated t-SNE plot. All DBSCAN results are combined into a cell similarity matrix. The cell similarity matrix is used to define ``CONSENSUS'' clusters conserved accross the previously defined clustering solutions. Identify marker genes for each concensus cluster.

Depends R (>= 4.1)

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.2

Suggests knitr, rmarkdown, BiocStyle, S4Vectors, matrixStats, dynamicTreeCut, testthat

VignetteBuilder knitr

Imports org.Hs.eg.db, org.Mm.eg.db, dbscan, fpc, factoextra, Biobase, BiocFileCache, parallel, doParallel, foreach, SummarizedExperiment, biomaRt, AnnotationDbi, methods, dplyr, scran, scater, pheatmap, ggplot2, gridExtra, SingleCellExperiment, stats, utils, scales, grDevices, graphics, Rtsne, GEOquery, clusterProfiler, stringr, tools, rlang, DT

biocViews Software, Technology, SingleCell, Sequencing, Clustering, ATACSeq, Classification

Collate 'AllClasses.R' 'AllGenerics.R' 'DataFormatting.R'
'checkFunctions.R' 'constructors.R' 'getters.R' 'globalVar.R'
'loadDataset.R' 'methods-clustering.R' 'methods-dbscan.R'
'methods-export.R' 'methods-markers.R'
'methods-normalization.R' 'sharedInternals.R' 'methods-plot.R'
'methods-tsne.R' 'runCONCLUS.R' 'setters.R'

git_url https://git.bioconductor.org/packages/conclus

git_branch RELEASE_3_14

git_last_commit 4f3d751

git_last_commit_date 2022-04-08

Date/Publication 2022-04-10

Author Ilyess Rachedi [cre],
Nicolas Descostes [aut],
Polina Pavlovich [aut],
Christophe Lancrin [aut]

Maintainer Ilyess Rachedi <rachedi.ilyess@gmail.com>

R topics documented:

`addClustering,scRNAseq-method`
addClustering

Description

This method enables to add a clustering to the existing object in order to change the coloration of the t-sne. It is particularly useful to compare the performance of different tools.

Usage

```
addClustering(theObject, filePathAdd=NA, clusToAdd=NA)
```

Arguments

`theObject`	An Object of class scRNASeq for which ?calculateClustersSimilarity was used.
`filePathAdd`	Default=NA. Path to the file containing the clustering to replace in the object. It should be made of two columns 'clusters' and 'cells'. This should be left to NA if clusToAdd is used. Default=NA.
`clusToAdd`	Data frame having two columns 'clusters' and 'cells' containing the clustering to be used in theObject. Should be left to NA if filePathAdd is used. Default=NA.

Value

An object of class scRNASeq with its column name metadata updated.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Retrieving the table indicating to which cluster each cell belongs
clustCellsDf <- retrieveTableClustersCells(scr)

## Replace "4" by "3" to merge 3/4
clustCellsDf$clusters[which(clustCellsDf$clusters == 4)] <- 3

## Modifying the object to take into account the new classification
scrUpdated <- addClustering(scr, clusToAdd=clustCellsDf)
```

`calculateClustersSimilarity,scRNAseq-method`
calculateClustersSimilarity

Description

Having computed cells similarity, pools information into clusters.

Usage

```
calculateClustersSimilarity(theObject, clusteringMethod = "ward.D2")
```

Arguments

`theObject`	An Object of class scRNASeq for which the count matrix was normalized (see ?normaliseCountMatrix), tSNE were calculated (see ?generateTSNECoordinates), dbScan was run (see ?runDBSCAN), and cells were clustered (see ?clusterCellsInternal).
`clusteringMethod`	Clustering method passed to hclust() function. See ?hclust for a list of method. Default = "ward.D2".

Value

An object of class scRNASeq with its clustersSimilarityMatrix and clustersSimiliratyOrdered slots updated.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

See Also

plotClustersSimilarity

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))
## Calculate clusters similarity
scr <- calculateClustersSimilarity(scr)
```

clusterCellsInternal,scRNAseq-method
clusterCellsInternal

Description

Returns consensus clusters by using hierarchical clustering on the similarity matrix of cells.

Usage

```
clusterCellsInternal(theObject, clusterNumber=NULL, deepSplit=4, cores=2,
                clusteringMethod="ward.D2")
```

Arguments

theObject	An Object of class scRNASeq for which the count matrix was normalized (see ?normaliseCountMatrix), tSNE were calculated (see ?generateTSNECoordinates), and dbScan was run (see ?runDBSCAN),
clusterNumber	Exact number of cluster. Default = NULL that will determine the number of clusters automatically.
deepSplit	Intuitive level of clustering depth. Options are 1, 2, 3, 4. Default = 4
cores	Maximum number of jobs that CONCLUS can run in parallel. Default is 1.
clusteringMethod	Clustering method passed to hclust() function. See ?hclust for a list of method. Default = "ward.D2".

Value

An object of class scRNASeq with its cellsSimilarityMatrix and sceNorm slots updated.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

See Also

plotCellSimilarity

Examples

```
## Object containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Compute the cell similarity matrix
scr <- clusterCellsInternal(scr, clusterNumber=4, cores=2)
```

conclusCacheClear *conclusCacheClear*

Description

This function deletes the cache of conclus.

Usage

```
conclusCacheClear()
```

Details

This function don't return anything. It deletes the current contents of the cache.

Value

Nothing, it deletes the cache of conclus.

Author(s)

Ilyess RACHEDI & Nicolas DESCOSTES

Examples

```
NULL
```

constructors *constructors*

Description

Constructor of the class scRNAseq.

Usage

```
singlecellRNAseq(experimentName, countMatrix, species, outputDirectory,
tSNElist=list(new("Tsne")), dbscanlist=list(new("Dbscan")),
cellSimMat= matrix(nrow = 1, ncol = 1, dimnames = list("c1", "c1"),
data = 1), clustSimMat=matrix(nrow = 1, ncol = 1, dimnames = list("1", "1"),
data = 1), clustSimOrdered=factor(1), markgenlist=list(data.frame(
Gene = c("gene1"), mean_log10_fdr = c(NA), n_05 = c(NA), score = c(NA))),
clustMark=data.frame(geneName="gene1", clusters=NA), genesInf = data.frame(
uniprot_gn_symbol=c("symbol"), clusters="1", external_gene_name="gene",
go_id="GO1,GO2", mgi_description="description",
entrezgene_description="descr", gene_biotype="gene", chromosome_name="1",
```

```
Symbol="symbol", ensembl_gene_id="ENS", mgi_id="MGI", entrezgene_id="1",
uniprot_gn_id="ID"))

TsneCluster(name, pc, perplexity, coordinates)

DbscanCluster(name, epsilon, minPoints, clustering)
```

Arguments

experimentName	Character string representing the name of the experiment.
countMatrix	An integer matrix representing the raw count matrix with reads or unique molecular identifiers (UMIs).
species	Character string representing the species of interest. Currently limited to "mouse" and "human". Other organisms can be added on demand.
outputDirectory	A character string of the path to the root output folder.
tSNElist	List of 'Tsne' objects representing the different tSNE coordinates generated by CONCLUS.
dbscanlist	List of 'Dbscan' objects representing the different Dbscan clustering generated by CONCLUS.
cellSimMat	A numeric Matrix defining how many times two cells have been associated to the same cluster across the 84 solutions (by default) of clustering.
clustSimMat	A numeric matrix comparing the robustness of the consensus clusters.
clustSimOrdered	A factor representing the clusters ordered by similarity.
markgenlist	List of data.frames. Each data frame contains the ranked genes of one cluster.
clustMark	A data frame containing the top 10 (by default) marker genes of each clusters.
genesInf	A data frame containing informations of the markers genes for each clusters.
name	A 'character' string representing the name of the Dbscan clustering.
pc	A 'numeric' value representing the number of principal components used by CONCLUS to perfom a PCA before calculating the tSNE.
perplexity	A 'numeric' vector. Default: c(30, 40)
coordinates	A 'numeric' matrix that contains the coordinates of one tSNE solution.
epsilon	A 'numeric' vector. The epsilon is the distance to consider two points belonging to the same cluster. Default = c(1.3, 1.4, 1.5)
minPoints	A 'numeric' value. The minPoints is the minimum number of points to construct a cluster.
clustering	A 'matrix' that contains the result of one DBSCAN clustering solution.

Value

Object of class scRNAseq

Object of class Tsne

Object of class Dbscan

See Also

scRNAseq-class

Tsne-class

Dbscan-class

Examples

```
experimentName <- "Bergiers"
outputDirectory <- "YourOutputDirectory"

## Load the count matrix
countmatrixPath <- system.file("extdata/countMatrix.tsv", package="conclus")
countMatrix <- loadDataOrMatrix(file=countmatrixPath, type="countMatrix",
                                ignoreCellNumber=TRUE)

## Load the coldata
coldataPath <- system.file("extdata/colData.tsv", package="conclus")
columnsMetaData <- loadDataOrMatrix(file=coldataPath, type="coldata",
                                    columnID="cell_ID")

## Create the initial object
scr <- singlecellRNAseq(experimentName = experimentName,
                countMatrix     = countMatrix,
                species         = "mouse",
                outputDirectory = outputDirectory)

mat <- matrix(seq_len(20), ncol=2)
colnames(mat) <- c("X", "Y")
TsneCluster(name = "test", pc = 30, perplexity = 4,
coordinates = mat)

DbscanCluster("test", 0.5, 2, matrix(1:10))
```

`Dbscan-class` *The Dbscan class*

Description

S4 class containing the features to plot DBSCAN. This constructor is internal and is used by the method runDBSCAN.

Slots

`name` A character string representing the name of the Dbscan clustering.

`epsilon` A numeric vector. The epsilon is the distance to consider two points belonging to the same cluster. Default = c(1.3, 1.4, 1.5).

`minPoints` A numeric value. The minPoints is the minimum number of points to construct a cluster.

`clustering` A matrix that contains the result of one DBSCAN clustering solution.

Constructor

Dbscan(name = "character", epsilon = "numeric", minPoints = "numeric", clustering = "matrix")

name: Empty character string or the name of the tSNE.
epsilon: Empty 'numeric' representing the epsilon.
minPoints: Empty 'numeric' representing the minPoints value.
clustering: Empty 'numeric' "matrix" or matrix of clustering.

Accessors

In the following snippets, x is a Dbscan object.

getName(x): Get the name of the Dbscan.
getEpsilon(x): Get the epsilon used.
getMinPoints(x): Get the MinPoint used.
getClustering(x): Get the matrix of DBSCAN clustering.

Subsetting

In the following snippets, x is a Dbscan object.

setName(x) <- value: Set the name of the Dbscan.
setEpsilon(x) <- value: Set the epsilon used.
setMinPoints(x) <- value: Set the minPoints used.
setClustering(x) <- value: Set the matrix of Dbscan clustering.

Author(s)

Ilyess Rachedi and Nicolas Descostes

See Also

runDBSCAN

`exportResults,scRNAseq-method` *exportResults*

Description

Export all the results of Conclus to a Results sub-directory.

Usage

```
exportResults(theObject, saveClusteringResults=TRUE, saveAll=FALSE,
saveNormalizedMatrix=FALSE, saveColData=FALSE, saveRowData=FALSE,
saveTsne=FALSE, saveDBScan=FALSE, saveCellsSimilarityMatrix=FALSE,
saveClustersSimilarityMatrix=FALSE, saveFullMarkers=FALSE,
saveTopMarkers=FALSE, saveGenesInfos=FALSE)
```

Arguments

theObject	An Object of class scRNASeq for which different steps of CONCLUS was applied to. The number of steps to run depends on what is wanted to be saved.
saveClusteringResults	Default=TRUE. Save the final clustering results giving the corresponding cluster number to each cell. The method ?calculateClustersSimilarity should have been run on the object. It is saved in the sub-directory 6_ConclusResult.
saveAll	Default=FALSE. Save all results of CONCLUS. The last step run on the scRNASeq object should be ?retrieveGenesInfo.
saveNormalizedMatrix	Default=FALSE. Save the normalized count matrix as a csv file. It is obtained with ?normaliseCountMatrix. The matrix is saved to the sub-directory '1_MatrixInfo'.
saveColData	Default=FALSE. Save the columns metadata of the normalized count matrix as a tsv file. These data are obtained with ?normaliseCountMatrix or were given as input of the method. These data are saved in the sub-directory '1_MatrixInfo'.
saveRowData	Default=FALSE. Save the raw metadata of the normalized count matrix as a tsv file. These data are obtained with ?normaliseCountMatrix. They are saved in the sub-directory '1_MatrixInfo'.
saveTsne	Default=FALSE. Save the tsne coordinates for each combination of PCs and perplexities as tsv files. These coordinates were obtained with ?generateTSNECoordinates. They are saved in the sub-directory '2_TSNECoordinates'.
saveDBScan	Default=FALSE. Save the clustering results of dbscan as tsv files. The number of clustering solutions is PCs*perplexity*epsilon*minPoints (see ?runDBSCAN, 84 solutions by default). These are saved in the sub-directory '3_dbScan'.
saveCellsSimilarityMatrix	Default=FALSE. Save the cells similarity matrix that was obtained with ?clusterCellsInternal. This matrix is saved in the sub-directory '4_CellSimilarityMatrix'.
saveClustersSimilarityMatrix	Default=FALSE. Save the cluster similarity matrix that was obtained with ?calculateClustersSimilarity. It is saved in the sub-directory '5_ClusterSimilarityMatrix'.
saveFullMarkers	Default=FALSE. Save the lists of markers that were obtained with ?rankGenes to the sub-directory 7_fullMarkers.

`saveTopMarkers` Default=FALSE. Save the top markers per clusters as csv files in the sub-directory '8_TopMarkers'. See ?retrieveTopClustersMarkers for more details.

`saveGenesInfos` Default=FALSE. Save the genes information for each cluster as csv files in the sub-directory '9_genesInfos'. See ?retrieveGenesInfo for more details.

Value

Sub-directories containing the results of the different conclus steps. See the 'Quick start' section of the vignette for details.

Author(s)

Ilyess RACHEDI and Nicolas DESCOSTES.

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Saving all results
exportResults(scr, saveAll=TRUE)

## Removing output directory
unlink("YourOutputDirectory", recursive=TRUE)
```

---

generateTSNECoordinates,scRNAseq-method
generateTSNECoordinates

Description

The function generates several t-SNE coordinates based on given perplexity and ranges of PCs. The final number of t-SNE plots is length(PCs)*length(perplexities).

Usage

```
generateTSNECoordinates(theObject, randomSeed=42, cores=2,
                PCs=c(4, 6, 8, 10, 20, 40, 50), perplexities=c(30,40),
                writeOutput = FALSE)
```

Arguments

`theObject` An Object of class scRNASeq for which the count matrix was normalized. See ?normaliseCountMatrix.

`randomSeed` Default is 42. Seeds used to generate the tSNE.

`cores` Maximum number of jobs that CONCLUS can run in parallel. Default is 1.

PCs	Vector of first principal components. For example, to take ranges 1:5 and 1:10 write c(5, 10). Default = c(4, 6, 8, 10, 20, 40, 50)
perplexities	A vector of perplexity (t-SNE parameter). See details. Default = c(30, 40)
writeOutput	If TRUE, write the tsne parameters to the output directory defined in theObject. Default = FALSE.

Details

Generates an object of fourteen (by default) tables with tSNE coordinates. Fourteen because it will vary seven values of principal components PCs=c(4, 6, 8, 10, 20, 40, 50) and two values of perplexity perplexities=c(30, 40) in all possible combinations. The chosen values of PCs and perplexities can be changed if necessary. We found that this combination works well for sc-RNA-seq datasets with 400-2000 cells. If you have 4000-9000 cells and expect more than 15 clusters, we recommend to take more first PCs and higher perplexity, for example, PCs=c(8, 10, 20, 40, 50, 80, 100) and perplexities=c(200, 240). For details about perplexities parameter see '?Rtsne'.

Value

An object of class scRNASeq with its tSNEList slot updated. Also writes coordinates in "dataDirectory/tsnes" subfolder if the parameter writeOutput is TRUE.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

See Also

normaliseCountMatrix

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Compute the tSNE coordinates
## These parameters are tweaked to fit our example data and reduce
## computing time, please consider using the default parameters or
## adjusted to your dataset.
scr <- generateTSNECoordinates(scr, perplexities=c(2,3),
                               PCs =c(4,5,6,7,8,9,10), cores=2)
```

getters *getters*

Description

Retrieve the data of the slots of a scRNA-seq, Tsne or Dbscan object.

Usage

```
getExperimentName(theObject)

getCountMatrix(theObject)

getSceNorm(theObject)

getSpecies(theObject)

getOutputDirectory(theObject)

getTSNEList(theObject)

getDbscanList(theObject)

getSuggestedClustersNumber(theObject)

getCellsSimilarityMatrix(theObject)

getClustersSimilarityMatrix(theObject)

getClustersSimilarityOrdered(theObject)

getMarkerGenesList(theObject, cluster)

getTopMarkers(theObject)

getGenesInfos(theObject)

## S4 method for signature 'Tsne'
getName(theObject)

getPerplexity(theObject)

getPC(theObject)

getCoordinates(theObject)

getName(theObject)
```

```
getEpsilon(theObject)

getMinPoints(theObject)

getClustering(theObject)
```

Arguments

theObject	A scRNA-seq, Tsne or Dbscan object. See description or ?scRNAseq, ?Tsne, ?Dbscan.
cluster	Integer, index of the cluster to retrive the gene list. If cluster="all", get the marker list for each cluster into a list.

Value

getExperimentName: Get the name of the experiment (scRNA-seq).

getCountMatrix: Get the count matrix (scRNA-seq).

getSceNorm: Get the SingleCellExperiment object used (scRNA-seq).

getSpecies: Get the species (scRNA-seq).

getOutputDirectory: Get the path of the output directory (scRNA-seq).

getTSNEList: Get the list of Tsne objects (scRNA-seq).

getDbscanList: Get the list of Dbscan objects (scRNA-seq).

getSuggestedClustersNumber: Get the suggested clusters number (scRNA-seq).

getCellsSimilarityMatrix: Get the cell similarity matrix (scRNA-seq).

getClustersSimilarityMatrix: Get the cluster similarity matrix (scRNA-seq).

getClustersSimilarityOrdered: Get the clusters ordered by similarity (scRNA-seq).

getMarkerGenesList: Get the list of marker genes by clusters (scRNA-seq).

getTopMarkers: Get the most significant markers by clusters (scRNA-seq).

getGenesInfos: Get informations about marker genes (scRNA-seq).

getPerplexity: Get the perplexity used (Tsne).

getPC: Get the PC used (Tsne).

getCoordinates: Get the matrix of tSNE coordinates (Tsne).

getName: Get the name of the tSNE or Dbscan object (Dbscan).

getEpsilon: Get the epsilon used (Dbscan).

getMinPoints: Get the MinPoint used (Dbscan).

getClustering: Get the matrix of DBSCAN clustering (Dbscan).

Author(s)

Ilyess RACHEDI

Examples

```
## Load the count matrix
countMatrixPath <- system.file("extdata/countMatrix.tsv", package="conclus")
countMatrix <- loadDataOrMatrix(file=countMatrixPath, type="countMatrix",
                                ignoreCellNumber=TRUE)

## Load the coldata
coldataPath <- system.file("extdata/colData.tsv", package="conclus")
columnsMetaData <- loadDataOrMatrix(file=coldataPath, type="coldata",
                                    columnID="cell_ID")

## Create the initial object
scr <- singlecellRNAseq(experimentName = "Bergiers",
                countMatrix     = countMatrix,
                species         = "mouse",
                outputDirectory = "YourOutputDirectory")

experimentName <- getExperimentName(scr)
countMatrix <- getCountMatrix(scr)
species <- getSpecies(scr)
outputDirectory <- getOutputDirectory(scr)
```

`loadDataOrMatrix` *loadDataOrMatrix*

Description

This function allows to import the coldata, rowData or the countMatrix. It formats each type of data to follow the requirements of CONCLUS.

Usage

```
loadDataOrMatrix(file, type, columnID=NULL, header=TRUE, sep='\t', dec=".",
                  ignoreCellNumber=FALSE)
```

Arguments

`file`	Path to the rowData, colData or Matrix.
`type`	Values should be "coldata", "rowdata", or "countMatrix".
`columnID`	For row and col data, column name containing cells/genes names/id. Should be the same names of genes or cells that in the count matrix. Should not be used when importing a matrix. Default=NULL.
`header`	Set TRUE if the first row of the table corresponds to the column names, and FALSE if it doesn't. Default=TRUE.
`sep`	Character used in the table to separate the fields. Usually it's ' ' ';' ',' or '\t'. Default='\t'.

dec Character used in the table for decimal points. Usually '.' or ',' . Default=".".

ignoreCellNumber
CONCLUS needs a large number of cells to collect statistics, so we recommend using CONCLUS if you have at least 100 cells. If you still want to use a count matrix with a number of cells inferior to 100 cells, set ignoreCellNumber=TRUE. Default=FALSE.

Value

The formatted row, col data or the matrix.

Author(s)

Ilyess RACHEDI and Nicolas DESCOSTES

Examples

```
## ColData
coldataPath <- system.file("extdata/colData.tsv", package="conclus")
loadDataOrMatrix(file=coldataPath, type="coldata", columnID="cell_ID")

## RowData
rowdataPath <- system.file("extdata/rowData.tsv",  package="conclus")
loadDataOrMatrix(file=rowdataPath, type="rowdata", columnID="gene_ID")

## Count matrix
countmatrixPath <- system.file("extdata/countMatrix.tsv", package="conclus")
countMatrix <- loadDataOrMatrix(file=countmatrixPath, type="countMatrix",
                                ignoreCellNumber=TRUE)
```

---

normaliseCountMatrix,scRNAseq-method

normaliseCountMatrix

Description

This function uses coldata (cells informations) and rowdata (genes informations) to filter the count matrix. It also normalizes by using deconvolution with size factors.

Usage

```
normaliseCountMatrix(theObject, sizes=c(20,40,60,80,100), rowdata=NULL,
                     coldata=NULL, alreadyCellFiltered=FALSE,
                     runQuickCluster=TRUE, info=TRUE, removeNoSymbol=FALSE)
```

Arguments

theObject	A scRNAseq object
sizes	Vector of size factors used by scran::computeSumFactors(). It is a numeric vector of pool sizes, i.e., number of cells per pool. See ?scran::computeSumFactors for more details.
rowdata	Data frame containing genes informations. Default is NULL.
coldata	Data frame containing cells informations. Default is NULL.
alreadyCellFiltered	Logical. If TRUE, quality check and filtering will not be applied.
runQuickCluster	Logical. If TRUE scran::quickCluster() function will be applied. It usually improves the normalization for medium-size count matrices. However, it is not recommended for datasets with less than 200 cells and may take too long for datasets with more than 10000 cells.
info	Logical. If TRUE, additional annotations like ensembl_gene_id, go_id, name_1006, chromosome_name and gene_biotype are added to the row data, for all the genes from the count matrix with ENSEMBL IDs or SYMBOL ID. Default: TRUE.
removeNoSymbol	Logical. If TRUE, genes with no SYMBOL are removed after the normalization

Details

This function uses the normalization method of the scater package. For more details about the normalization used see ?scater::normalize. The size factors used in the normalization are calculated with scran::computeSumFactors.

Beforehand, the function will annotate genes creating rowData and add statistics about cells into columnsMetaData. If you already have columnsMetaData and rowData, you can give it to the function (see manual). It will keep your columns and add new ones at the end. If you do not want to lose any cell after quality metrics check, select alreadyCellFiltered = TRUE, by default it is FALSE. Before scater normalization, the function will call scran::quickCluster (see manual for details). If you want to skip this step, set runQuickCluster = FALSE, by default it is TRUE. We advice to first try the analysis with this option and to set it to FALSE if no rare populations are found.

Value

Returns a scRNASeq object with its sceNorm slot updated. This slot contains a SingleCellExperiment object having the normalized count matrix, the colData (table with cells informations), and the rowData (table with the genes informations). See ?SingleCellExperiment for more details.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

Examples

```
## Load the count matrix
countmatrixPath <- system.file("extdata/countMatrix.tsv", package="conclus")
countMatrix <- loadDataOrMatrix(file=countmatrixPath, type="countMatrix",
```

```
                                ignoreCellNumber=TRUE)

## Load the coldata
coldataPath <- system.file("extdata/colData.tsv", package="conclus")
columnsMetaData <- loadDataOrMatrix(file=coldataPath, type="coldata",
columnID="cell_ID")

## Create the initial object
scr <- singlecellRNAseq(experimentName = "Bergiers",
                countMatrix     = countMatrix,
                species         = "mouse",
                outputDirectory = "YourOutputDirectory")

## Normalize and filter the raw counts matrix
scr <- normaliseCountMatrix(scr, coldata = columnsMetaData, info=FALSE)
```

`plotCellHeatmap,scRNAseq-method`
plotCellHeatmap

Description

This function plots heatmap with marker genes on rows and clustered cells on columns.

Usage

```
plotCellHeatmap(theObject, fileName = NA, meanCentered=TRUE,
                colorPalette="default", statePalette="default",
                clusteringMethod="ward.D2", orderClusters=FALSE,
                orderGenes=FALSE, returnPlot=FALSE, savePlot=FALSE, width=10,
                height=8.5, onefile=FALSE, clusterCols=FALSE,
                showColnames=FALSE, fontsize=7.5,  fontsizeRow=8,
                plotPDF=TRUE, widthPNG=800, heightPNG=750, silentPlot)
```

Arguments

`theObject`	A scRNAseq object with the cluster similarity matrix obtained with ?calculateClustersSimilarity method and the top markers obtained with ?retrieveTopClustersMarkers.
`fileName`	Name of the output file to which the heatmap is saved.
`meanCentered`	Boolean indicating if mean centering should be applied to the expression matrix. Default = TRUE.
`colorPalette`	A vector of colors for clusters. Default = "default", See details.
`statePalette`	A vector of colors for states or conditions. See details.
`clusteringMethod`	Clustering method passed to hclust() function. See ?hclust for a list of method. Default = "ward.D2".

orderClusters	If True, clusters in the similarity matrix of cells will be ordered by name. Default = FALSE.
orderGenes	Boolean, should the heatmap be structured by gene. Default = FALSE.
returnPlot	If TRUE returns a pheatmap object. Default=FALSE.
savePlot	If TRUE and plotPDF=TRUE, save the heatmap in pdf format. The heatmap is saved in the output directory defined in theObject (?getOutputDirectory) and in the sub-directory 'pictures'.
width	Width of the plot in the pdf file. See ?pdf for more details. Default = 10.
height	Height of the plot in the pdf file. See ?pdf for more details. Default = 8.5.
onefile	Logical: if TRUE allow multiple figures in one file. If FALSE, generate a file with name containing the page number for each page. Defaults to FALSE.
clusterCols	If TRUE, the columns representing the clusters are also taken into account in the hierarchical clustering. Default=FALSE.
showColnames	Shoud the names of the columns (clusters) be indicated on the heatmap. Default = FALSE.
fontsize	Base fontsize for the plot. Default = 7.5.
fontsizeRow	Fontsize for rownames. Default = 8.
plotPDF	If TRUE, the heatmap is saved in pdf format and in png otherwise. Default = TRUE.
widthPNG	Width of the png. See ?png for details. Default=800.
heightPNG	Height of the png. See ?png for details. Default=750.
silentPlot	If TRUE, does not plot the pheatmap. Default=FALSE.

Details

colorPalette/statePalette – A vector of colors for clusters/states or 'default' value. If 'default' is selected, the number of clusters is limited to 16. If an error message is thrown, re-run the function with your own color vector.

Value

A pheatmap object of the heatmap if returnPlot is TRUE.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

See Also

calculateClustersSimilarity plotClusteredTSNE plotCellSimilarity plotGeneExpression plotClustersSimilarity

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Plot the heatmap with marker genes
plotCellHeatmap(scr)
```

`plotCellSimilarity,scRNAseq-method`
plotCellSimilarity

Description

This function plots the similarity matrix as a heatmap.

Usage

```
plotCellSimilarity(theObject, colorPalette="default",
            statePalette="default", clusteringMethod="ward.D2",
            orderClusters=FALSE, savePlot=FALSE, plotPDF=TRUE,
            returnPlot=FALSE, width=7, height=6, onefile=FALSE,
            showRowNames=FALSE, showColnames=FALSE, fontsize=7.5,
            fontsizeRow=0.03, widthPNG=800, heightPNG=750, silentPlot=FALSE)
```

Arguments

`theObject`	An Object of class scRNASeq for which the count matrix was normalized (see ?normaliseCountMatrix), tSNE were calculated (see ?generateTSNECoordinates), dbScan was run (see ?runDBSCAN), cells were clustered (see ?clusterCellsInternal), as clusters themselves (see ?calculateClustersSimilarity).
`colorPalette`	A vector of colors for clusters. Default = "default", see details.
`statePalette`	A vector of colors for states or conditions. See details.
`clusteringMethod`	Clustering method passed to hclust() function. See ?hclust for a list of method. Default = "ward.D2"
`orderClusters`	If TRUE, clusters in the similarity matrix of cells will be ordered by name. Default = FALSE.
`savePlot`	If TRUE, the heatmap is saved in the directory defined in theObject (?getOutputDirectory) and in the sub-directory "pictures".
`plotPDF`	If TRUE export heatmap in pdf format, if FALSE export it in png format. Default=TRUE.
`returnPlot`	Boolean indicating if the pHeatmap object should be returned by the function. Default = TRUE.
`width`	Width of the plot in the pdf file. See ?pdf for more details. Default = 7.

height	Height of the plot in the pdf file. See ?pdf for more details. Default = 6.
onefile	Logical: if TRUE allow multiple figures in one file. If FALSE, generate a file with name containing the page number for each page. Defaults to FALSE.
showRowNames	pheatmap parameter. Boolean specifying if row names are displayed. Default=FALSE.
showColnames	pheatmap parameter. Boolean specifying if column names are displayed. Default=FALSE.
fontsize	pheatmap parameter. Base fontsize for the plot. Default=7.5.
fontsizeRow	pheatmap parameter. Fontsize for rownames. Default=0.03.
widthPNG	Width of the png. See ?png for details. Default=800.
heightPNG	Height of the png. See ?png for details. Default=750.
silentPlot	If TRUE, does not plot the heatmap. Default=FALSE.

Details

colorPalette/statePalette – A vector of colors for clusters/states or 'default' value. If 'default' is selected, the number of clusters is limited to 16. If an error message is thrown, re-run the function with your own color vector.

Value

A pheatmap object of the similarity heatmap if returnPlot is TRUE.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

See Also

calculateClustersSimilarity plotClusteredTSNE plotCellHeatmap plotGeneExpression plotClustersSimilarity

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Plot the heatmap of the similarity matrix
plotCellSimilarity(scr)
```

`plotClusteredTSNE,scRNAseq-method`
plotClusteredTSNE

Description

Plot t-SNE generated with different PCs and perplexities. It can also use a coloring scheme by clusters or states. The latter is possible if a 'state' column, representing conditions, is provided in the columns meta-data.

Usage

```
plotClusteredTSNE(theObject, colorPalette="default",
            PCs=c(4, 6, 8, 10, 20, 40, 50), perplexities=c(30, 40),
            columnName="clusters", savePlot=FALSE, plotPDF=TRUE,
            returnPlot=FALSE, width=6, height=5, onefile=FALSE, widthPNG=800,
            heightPNG=750, silentPlot=FALSE, tSNENb=NA)
```

Arguments

`theObject`	An Object of class scRNASeq for which the count matrix was normalized (see ?normaliseCountMatrix), tSNE were calculated (see ?generateTSNECoordinates), dbScan was run (see ?runDBSCAN), cells were clustered (see ?clusterCellsInternal), as clusters themselves (see ?calculateClustersSimilarity).
`colorPalette`	A vector of colors for clusters. Default = "default", see details.
`PCs`	A vector of first principal components. For example, to take ranges 1:5 and 1:10 write c(5, 10). Default = c(4, 6, 8, 10, 20, 40, 50). See ?generateTSNECoordinates for details.
`perplexities`	Numeric scalar defining the perplexity parameter, see ?Rtsne and ?generateTSNECoordinates for more details. Default = c(30, 40)
`columnName`	Name of the column to color the t-SNE with. Possible values are clusters, noColor, or state.
`savePlot`	If TRUE, the heatmap is saved in the directory defined in theObject (?getOutputDirectory) and in the sub-directory "pictures/tSNE_pictures".
`plotPDF`	If TRUE export heatmap in pdf format, if FALSE export it in png format. Default=TRUE.
`returnPlot`	Boolean indicating if the pHeatmap object should be returned by the function. Default = FALSE.
`width`	Width of the plot in the pdf file. See ?pdf for more details. Default = 6.
`height`	Height of the plot in the pdf file. See ?pdf for more details. Default = 5.
`onefile`	Logical: if TRUE allow multiple figures in one file. If FALSE, generate a file with name containing the page number for each page. Defaults to FALSE.
`widthPNG`	Width of the png. See ?png for details. Default=800.

heightPNG	Height of the png. See ?png for details. Default=750.
silentPlot	If TRUE, the plots are not displayed on the current device. Default=FALSE.
tSNENb	Give the number of the tSNE to plot. If NA, all tSNE solutions are plotted (14 tSNE by default). Default=NA.

Details

colorPalette – A vector of colors for clusters/states or 'default' value. If 'default' is selected, the number of clusters is limited to 16. If an error message is thrown, re-run the function with your own color vector.

Value

A list of ggplot objects if returnPlot is TRUE.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

See Also

calculateClustersSimilarity plotCellSimilarity plotCellHeatmap plotGeneExpression plotClustersSimilarity

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Plot the heatmap of the similarity matrix
plotClusteredTSNE(scr, tSNENb=1)
```

---

plotClustersSimilarity,scRNAseq-method
plotClustersSimilarity

Description

This function plots the clusters similarity matrix as a heatmap.

Usage

```
plotClustersSimilarity(theObject, colorPalette="default",
statePalette="default", clusteringMethod="ward.D2", returnPlot=FALSE,
savePlot=FALSE, plotPDF=TRUE, width=7, height=5.5, onefile=FALSE,
fontsize=7.5, widthPNG=800, heightPNG=750, silentPlot=FALSE)
```

Arguments

theObject	A scRNAseq object with the cluster similarity matrix obtained with ?calculateClustersSimilarity.
colorPalette	A vector of colors for clusters. Default = "default", see details.
statePalette	A vector of colors for states or conditions. See details.
clusteringMethod	Clustering method passed to hclust() function. See ?hclust for a list of method. Default = "ward.D2"
returnPlot	Boolean indicating if the pHeatmap object should be returned by the function. Default = FALSE.
savePlot	If TRUE and plotPDF=TRUE, save the heatmap in pdf format. The heatmap is saved in the output directory defined in theObject (?getOutputDirectory) and in the sub-directory 'pictures'.
plotPDF	If TRUE, the heatmap is saved in pdf format and in png otherwise. Default = TRUE.
width	Width of the plot in the pdf file. See ?pdf for more details. Default = 7.
height	Height of the plot in the pdf file. See ?pdf for more details. Default = 5.5.
onefile	Logical: if TRUE allow multiple figures in one file. If FALSE, generate a file with name containing the page number for each page. Defaults to FALSE.
fontsize	pheatmap parameter. Base fontsize for the plot. Default=7.5.
widthPNG	Width of the png. See ?png for details. Default=800.
heightPNG	Height of the png. See ?png for details. Default=750.
silentPlot	If TRUE, does not plot the pheatmap. Default=FALSE. This is useful if one wants to only retrieve the object to insert the figure in a grid for instance.

Details

colorPalette/statePalette – A vector of colors for clusters/states or 'default' value. If 'default' is selected, the number of clusters is limited to 16. If an error message is thrown, re-run the function with your own color vector.

Value

A pheatmap object of the clusters similarity matrix.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

See Also

calculateClustersSimilarity plotClusteredTSNE plotCellHeatmap plotGeneExpression plotCellSimilarity

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Plot similarity matrix as a heatmap
plotClustersSimilarity(scr)
```

`plotGeneExpression,scRNAseq-method`
plotGeneExpression

Description

The function saves a t-SNE plot colored by expression of a given gene.

Usage

```
plotGeneExpression(theObject, geneName,
palette=c("grey","red", "#7a0f09", "black"), returnPlot=FALSE,
            tSNEpicture=1, savePlot=FALSE, alpha=1, limits=NA,
            pointSize=1, width=6, height=5, plotPDF=TRUE, silentPlot=FALSE)
```

Arguments

`theObject`	A scRNAseq object with the top markers retrieved. See ?retrieveTopClusters-Markers.
`geneName`	Name of the gene to highlight on the t-SNE plot.
`palette`	Color palette for the expression levels.
`returnPlot`	If TRUE, returns a ggplot object of the tSNE. Default = FALSE.
`tSNEpicture`	Number of the tSNE picture that you want to use for plotting the gene expression. Default = 1.
`savePlot`	If TRUE, save the tSNE in pdf or png format. Default=FALSE.
`alpha`	Opacity of the points of the plot. Default = 1.
`limits`	Range of the gene expression shown in the legend. Default = NA. See details.
`pointSize`	Size of the points on the tSNE. Default = 1.
`width`	Width of the plot. Default = 6.
`height`	Height of the plot. Default = 5.
`plotPDF`	If TRUE export tSNE in pdf format, if FALSE export it in png format. Default=TRUE.
`silentPlot`	If TRUE, the plots are not displayed on the current device. Default=FALSE. This is useful if one wants to only retrieve the object to insert the figure in a grid for instance.

Details

limits – This option allows generating t-SNE plots with equal color scale to compare the expression of different genes. By default, limits are the range of expression of a selected gene.

Value

A ggplot object of the gene expression colored tSNE.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

See Also

retrieveTopClustersMarkers plotCellSimilarity plotCellHeatmap plotClusteredTSNE plotClustersSimilarity

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## t-SNE plot colored by expression of a  given gene.
plotGeneExpression(scr, getTopMarkers(scr)[1,1])
```

---

rankGenes,scRNAseq-method *rankGenes*

Description

This function searches marker genes for each cluster.

Usage

```
rankGenes(theObject, column="clusters", writeMarkerGenes=FALSE)
```

Arguments

theObject	An Object of class scRNASeq for which the count matrix was normalized (see ?normaliseCountMatrix), tSNE were calculated (see ?generateTSNECoordinates), dbScan was run (see ?runDBSCAN), cells were clustered (see ?clusterCellsInternal), as clusters themselves (see ?calculateClustersSimilarity).
column	Name of the column with a clustering result. Default="clusters"
writeMarkerGenes	If TRUE, output one list of marker genes per cluster in the output directory defined in theObject and in the sub-directory 'marker_genes'. Default=FALSE.

Details

To understand the nature of the consensus clusters identified by CONCLUS, it is essential to identify genes which could be classified as marker genes for each cluster. To this aim, each gene should be "associated" to a particular cluster. This association is performed by looking at upregulated genes in a particular cluster compared to the others (multiple comparisons).

The function rankGenes performs multiple comparisons of all genes from theObject and rank them according to a score reflecting a FDR power.

For each table corresponding to a particular consensus cluster, the first column is a gene name. The following columns represent adjusted p-values (FDR) of a one-tailed T-test between the considered cluster and all others.

Top genes with significant FDR in most of the comparisons can be assumed as positive markers of a cluster. The column mean_log10_fdr is the mean power of FDR in all comparisons; the column n_05 is the number of comparisons in which the gene was significantly upregulated. The score for marker genes is the average power of FDR among all comparisons for a cluster multiplied to weights taken from the clustersSimilarityMatrix + 0.05. Taking into account both FDRs of all comparisons and clustersSimilarityMatrix allows us to keep the balance between highlighting markers for individual clusters and their 'families' which makes the final heatmap as informative as possible.

Note: Adding 0.05 to the clustersSimilarityMatrix in calculating the score helps avoiding the following problem: in case you have a cluster very different from all others, it will have the value 1 on the diagonal and 0 similarities to all others groups in the clustersSimilarityMatrix. So all weights for that cluster will be zeros meaning that the score would also be zero and genes will be ordered in alphabetical order in the corresponding marker genes list file.

For a cluster k and a gene G, a scoreG was defined in the following way:

scoreG= sum((-log10(fdrk, i + epsilon)*weightk,i) / nClusters-1)

Where

1. fdrk,i is an adjusted p-value obtained by comparing expression of G in cluster k versus expression of G in cluster i.
2. weightk,i is a similarity between these two groups taken from the element in the clustersSimilarityMatrix.
3. nClusters is a number of consensus clusters given to the rankGenes().
4. epsilon = 10-300 is a small number which does not influence the ranking and added to avoid an error when fdr is equal to zero.
5. k = [1,. . . ,nClusters].
6. I = ([1,. . . ,nClusters]exceptfor[k]).

Value

An object of class scRNASeq with its markerGenesList slot updated.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

See Also

retrieveTopClustersMarkers retrieveGenesInfo

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Ranking genes
scr <- rankGenes(scr)
```

`retrieveFromGEO` *retrieveFromGEO*

Description

This function retrieves the count matrix and columns meta-data from GEO. They are formatted to be suitable inputs for conclus.

Usage

```
retrieveFromGEO(matrixURL, countMatrixPath, species,
seriesMatrixName=NA, metaDataPath=NA, colMetaDataURL=NA,
convertToSymbols=TRUE, annoType="ENSEMBL")
```

Arguments

`matrixURL`	URL of the count matrix. The matrix must be un-normalized.
`countMatrixPath`	Path to the file to which the downloaded count matrix will be saved.
`species`	Values should be 'mouse' or 'human'. Other organisms can be added on demand.
`seriesMatrixName`	Name of the columns meta-data file hosted on GEO. This name can usually be found in the 'Series Matrix File(s)' section. Should not be used if colMetaDataURL is defined. Default=NA.
`metaDataPath`	If colMetaDataURL is used, defines the path to the file to which the downloaded meta-data will be saved.
`colMetaDataURL`	URL of the columns meta-data file hosted on GEO. This file can be found in 'supplementary file'. Should not be used if seriesMatrixName is defined. Default=NA.
`convertToSymbols`	Boolean indicating if the genes IDs contained in the row names of the matrix should be converted to official genes symbols. Default: TRUE. To choose the type of IDs contained in the count matrix, see the annoType parameter just below.
`annoType`	Type of the genes annotations contained in the row names of the count Matrix. Default: "ENSEMBL". See details.

Details

The conversion (TRUE by default) of the row genes IDs (ENSEMBL by default) to official genes symbols is done with the function 'bitr' of the 'clusterProfiler' package. To see a list of all possible values to pass to the annoType parameter use 'keytypes' method on "org.Mm.eg.db" (for mouse) or "org.Hs.eg.db" (for human). For example, copy/paste in a R terminal: library(org.Mm.eg.db);keytypes(org.Mm.eg.db)

Value

A list. The first element contains the count matrix and the second element contains the columns meta-data.

Author(s)

Nicolas DESCOSTES & Ilyess RACHEDI

Examples

```
outputDirectory <- "./YourOutputDirectory"
dir.create(outputDirectory, showWarnings=FALSE)
species <- "mouse"

countMatrixPath <- file.path(outputDirectory, "countmatrix.txt")
matrixURL <- paste0("https://www.ncbi.nlm.nih.gov/geo/download/?acc=",
"GSE96982&format=file&file=GSE96982%5FcountMatrix%2Etxt%2Egz")
seriesMatrix <- "GSE96982-GPL19057_series_matrix.txt.gz"

result <- retrieveFromGEO(matrixURL, countMatrixPath, species,
seriesMatrixName=seriesMatrix)

countMatrix <- result[[1]]
columnsMetaData <- result[[2]]
```

---

retrieveGenesInfo,scRNAseq-method

retrieveGenesInfo

Description

This method retrieve information about the marker genes of each cluster querying the Ensembl database with biomaRt and display the result.

Usage

```
retrieveGenesInfo(theObject, groupBy="clusters",
                orderGenes="initial", getUniprot=TRUE, cores=2,
                saveInfos=FALSE)
```

Arguments

theObject	An Object of class scRNASeq for which the count matrix was normalized (see ?normaliseCountMatrix), tSNE were calculated (see ?generateTSNECoordinates), dbScan was run (see ?runDBSCAN), cells were clustered (see ?clusterCellsInternal), as clusters themselves (see ?calculateClustersSimilarity), and ?rankGenes as ?retrieveTopMarkers.
groupBy	A column in the input table used for grouping the genes in the output tables. This option is useful if a table contains genes from different clusters. Default = "clusters".
orderGenes	If "initial" then the order of genes will not be changed. The other option is "alphabetical". Default = "initial".
getUniprot	Boolean, whether to get information from UniProt or not. Default = TRUE.
cores	Maximum number of jobs that CONCLUS can run in parallel. Default is 1.
saveInfos	If TRUE, save the genes infos table in the directory defined in theObject (?getOutputDirectory), in the sub-directory 'marker_genes/saveGenesInfo'.

Details

The output dataframe is composed of the following columns:

- uniprot_gn_symbol: Uniprot gene symbol.
- clusters: The cluster to which the gene is associated.
- external_gene_name: The complete gene name.
- go_id: Gene Ontology (GO) identification number.
- mgi_description: If the species is mouse, description of the gene on MGI.
- entrezgene_description: Description of the gene by Entrez database.
- gene_biotype: protein coding gene, lincRNA gene, miRNA gene, unclassified non-coding RNA gene, or pseudogene.
- chromosome_name: The chromosome on which the gene is located.
- Symbol: Official gene symbol.
- ensembl_gene_id: ID of the gene on the ensembl database.
- mgi_id: If the species is mouse, ID of the gene on the MGI database.
- entrezgene_id: ID of the gene on the entrez database.
- uniprot_gn_id: ID of the gene on the uniprot database.

Value

Display a table with the info retrieved.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

See Also

rankGenes retrieveTopClustersMarkers

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Getting genes info
scr <- retrieveGenesInfo(scr, cores=2)
```

---

retrieveTableClustersCells,scRNAseq-method
retrieveTableClustersCells

Description

Having computed clusterCellsInternal, retrieve to what cluster each cell belongs. The output data.frame can be passed to the method ?addClustering.

Usage

```
retrieveTableClustersCells(theObject)
```

Arguments

theObject	An Object of class scRNASeq for which the cells were clustered internally. See ?clusterCellsInternal.

Value

A data frame containing two columns 'clusters' and 'cells' indicating the result of the consensus clustering at the cellular level.

Author(s)

Nicolas DESCOSTES.

See Also

addClustering

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Retrieving the table clusters-cells.
cellClustDf <- retrieveTableClustersCells(scr)
```

`retrieveTopClustersMarkers,scRNAseq-method`
retrieveTopClustersMarkers

Description

This function retrieves the top N marker genes for each cluster.

Usage

```
retrieveTopClustersMarkers(theObject, nTop=10, removeDuplicates = TRUE,
                writeMarkerGenes = FALSE)
```

Arguments

`theObject`	An Object of class scRNASeq for which rankGenes was run. See ?rankGenes.
`nTop`	Number of marker genes to retrieve per cluster. Default=10.
`removeDuplicates`	If TRUE, duplicated markers are removed from the lists. Default=TRUE.
`writeMarkerGenes`	If TRUE, writes one list per cluster in the output folder defined in theObject, and in the sub-directory marker_genes/markers_lists. Default=FALSE.

Value

Output the list of markers to marker_genes/markers_lists if writeMarkersGenes is TRUE and return a scRNASeq object with its clustersMarkers slot updated.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

See Also

retrieveGenesInfo

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Retrieve the top 10 markers per cluster
scr <- retrieveTopClustersMarkers(scr)
```

runCONCLUS *runCONCLUS*

Description

This function is a wrapper to run the whole CONCLUS workflow. See details.

Usage

```
runCONCLUS(
        ## General parameters
        outputDirectory, experimentName, countMatrix, species, cores=2,
        clusteringMethod="ward.D2", exportAllResults=TRUE,
        orderClusters=FALSE, clusToAdd=NA, silentPlot=TRUE,

        ## Normalisation parameters
        sizes=c(20,40,60,80,100), rowMetaData=NULL, columnsMetaData = NULL,
        alreadyCellFiltered=FALSE, runQuickCluster=TRUE, info=TRUE,

        ## tSNE parameters
        randomSeed = 42, PCs=c(4, 6, 8, 10, 20, 40, 50),
        perplexities=c(30,40), writeOutputTSne = FALSE,

        ## Dbscan parameters
        epsilon=c(1.3, 1.4, 1.5), minPoints=c(3, 4), writeOutputDbScan=FALSE,

        ## Cell Similarity matrix parameters
        clusterNumber=10, deepSplit=4,

        ## Rank genes parameters
        columnRankGenes="clusters", writeOutputRankGenes=FALSE,

        ## Retrieving top markers parameters
        nTopMarkers=10, removeDuplicates = TRUE, writeTopMarkers=FALSE,

        ## Retrieving genes infos parameters
        groupBy="clusters", orderGenes="initial", getUniprot=TRUE,
        saveInfos=FALSE,

        ## plotCellSimilarity parameters
        colorPalette="default", statePalette="default", writeCSM=FALSE,
        widthCSM=7, heightCSM=6,

        ## plotClusteredTSNE parameters
        savePlotCTSNE=FALSE, widthPlotClustTSNE=6, heightPlotClustTSNE=5,
        tSNENb=NA,
```

```
## plotCellHeatmap parameters
meanCentered=TRUE, orderGenesCH=FALSE, savePlotCH=FALSE, widthCH=10,
heightCH=8.5, clusterCols=FALSE,

## plotClustersSimilarity parameters
savePlotClustSM=FALSE, widthPlotClustSM=7, heightPlotClustSM=5.5)
```

Arguments

outputDirectory	Directory to which results should be written. This needs to be defined even if you choose to not output any results.
experimentName	String of the name of the experiment.
countMatrix	Matrix containing the raw counts.
species	Character string of the species of interest. Shoud be mouse or human. Other organisms can be added on demand.
cores	Maximum number of jobs that CONCLUS can run in parallel. This parameter is used by ?generateTSNECoordinates, ?runDBSCAN, ?clusterCellsInternal, and ?retrieveGenesInfo. Default=1.
clusteringMethod	Clustering method passed to hclust() function. See ?hclust for a list of method. This parameter is used by ?clusterCellsInternal, ?calculateClustersSimilarity, ?plotCellSimilarity, ?plotClusteredTSNE, ?plotCellHeatmap, and ?plotClustersSimilarity. Default = "ward.D2".
exportAllResults	If TRUE, Save all results of CONCLUS. See ?exportResults for details. Default=TRUE.
orderClusters	If TRUE, clusters in the cells and clusters similarity matrix of cells will be ordered by name. Default = FALSE.
clusToAdd	If not NA, defines the clustering to be used in theObject. This is particularly useful when one wants to compare the clustering performance of different tools. It should be a data frame having two columns 'clusters' and 'cells'. Default=NA.
silentPlot	Boolean indicating if the figures should not be output on the R graphics. Default=TRUE.
sizes	Vector of size factors from scran::computeSumFactors() function used by ?normaliseCountMatrix.
rowMetaData	Data frame containing genes informations. Default is NULL. See ?normaliseCountMatrix.
columnsMetaData	Data frame containing cells informations. Default is NULL. See ?normaliseCountMatrix.
alreadyCellFiltered	If TRUE, quality check and filtering will not be applied during the normalization of the count matrix. See ?normaliseCountMatrix.

`runQuickCluster`
: If TRUE scran::quickCluster() function will be applied. It usually improves the normalization for medium-size count matrices. However, it is not recommended for datasets with less than 200 cells and may take too long for datasets with more than 10000 cells. Default=TRUE. See ?normaliseCountMatrix.

`info`
: Logical. If TRUE, additional annotations like ensembl_gene_id, go_id, name_1006, chromosome_name and gene_biotype are added to the row data, for all the genes from the count matrix with ENSEMBL IDs or SYMBOL ID. Default: TRUE.

`randomSeed`
: Default is 42. Seeds used to generate the tSNE. See ?generateTSNECoordinates.

`PCs`
: Vector of first principal components. For example, to take ranges 1:5 and 1:10 write c(5, 10). Default = c(4, 6, 8, 10, 20, 40, 50). See ?generateTSNECoordinates.

`perplexities`
: A vector of perplexity (t-SNE parameter). See ?generateTSNECoordinates for details. Default = c(30, 40).

`writeOutputTSne`
: If TRUE, write the tsne parameters to the output directory defined in theObject. Default = FALSE. Ignored if exportAllResults=TRUE.

`epsilon`
: Reachability distance parameter of fpc::dbscan() function. See Ester et al. (1996) for more details. Default = c(1.3, 1.4, 1.5).

`minPoints`
: Reachability minimum no. of points parameter of fpc::dbscan() function. See Ester et al. (1996) for more details. Default = c(3, 4).

`writeOutputDbScan`
: If TRUE, write the results of the dbScan clustering to the output directory defined in theObject, in the sub-directory output_tables. Default = FALSE. Ignored if exportAllResults=TRUE.

`clusterNumber`
: Exact number of cluster. Default = NULL that will determine the number of clusters automatically. See ?clusterCellsInternal.

`deepSplit`
: Intuitive level of clustering depth. Options are 1, 2, 3, 4. See ?clusterCellsInternal. Default = 4.

`columnRankGenes`
: Name of the column with a clustering result. See ?rankGenes. Default="clusters".

`writeOutputRankGenes`
: If TRUE, output one list of marker genes per cluster in the output directory defined in theObject and in the sub-directory 'marker_genes'. Default=FALSE. Ignored if exportAllResults=TRUE.

`nTopMarkers`
: Number of marker genes to retrieve per cluster. See ?retrieveTopClustersMarkers. Default=10.

`removeDuplicates`
: If TRUE, duplicated markers are removed from the lists. See ?retrieveTopClustersMarkers. Default=TRUE.

`writeTopMarkers`
: If TRUE, writes one list per cluster in the output folder defined in theObject, and in the sub-directory marker_genes/markers_lists. Default=FALSE. Ignored if exportAllResults=TRUE.

groupBy A column in the input table used for grouping the genes in the output tables. This option is useful if a table contains genes from different clusters. See ?retrieveGenesInfo. Default = "clusters".

orderGenes If "initial" then the order of genes will not be changed. The other option is "alphabetical". See ?retrieveGenesInfo. Default="initial".

getUniprot Boolean, whether to get information from UniProt or not. See ?retrieveGenesInfo. Default = TRUE.

saveInfos If TRUE, save the genes infos table in the directory defined in theObject (?getOutputDirectory) and in the sub-directory 'marker_genes/saveGenesInfo'. Default=FALSE. Ignored if exportAllResults=TRUE.

colorPalette A vector of colors for clusters. This parameter is used by all plotting methods. Default = "default". See ?plotClustersSimilarity for details.

statePalette A vector of colors for states or conditions. This parameter is used by all plotting functions except ?plotClusteredTSNE. See ?plotClustersSimilarity for details.

writeCSM If TRUE, the cells similarity heatmap is saved in the directory defined in theObject (?getOutputDirectory) and in the sub-directory "pictures". Default=FALSE. Ignored if exportAllResults=TRUE.

widthCSM Width of the plot in the pdf file. See ?pdf for more details. Default = 7.

heightCSM Height of the plot in the pdf file. See ?pdf for more details. Default = 6.

savePlotCTSNE If TRUE, the heatmap of the clustered tSNE is saved in the directory defined in theObject (?getOutputDirectory) and in the sub-directory "pictures/tSNE_pictures". Default=FALSE. Ignored if exportAllResults=TRUE.

widthPlotClustTSNE
Width of the clustered tSNE plot in the pdf file. See ?pdf for more details. Default = 6.

heightPlotClustTSNE
Height of the clustered tSNE plot in the pdf file. See ?pdf for more details. Default = 5.

tSNENb Give the number of the tSNE to plot. If NA, all tSNE solutions are plotted (14 tSNE by default). Default=NA.

meanCentered Boolean indicating if mean centering should be applied to the expression matrix. See ?plotCellHeatmap. Default = TRUE.

orderGenesCH Boolean, should the heatmap be structured by gene. See ?plotCellHeatmap. Default=FALSE.

savePlotCH If TRUE save the cell heatmap in pdf format. The heatmap is saved in the output directory defined in theObject (?getOutputDirectory) and in the sub-directory 'pictures'. Default=FALSE. Ignored if exportAllResults=TRUE.

widthCH Width of the cell heatmap saved in ?pdf. Default = 10.

heightCH Height of the cell heatmap saved in ?pdf. Default = 8.5.

clusterCols If TRUE, the columns representing the clusters are also taken into account in the hierarchical clustering of the cell heatmap. Default=FALSE.

savePlotClustSM
If TRUE, save the cluster similarity heatmap in pdf format. The heatmap is saved in the output directory defined in theObject (?getOutputDirectory) and in the sub-directory 'pictures'. Default=FALSE. Ignored if exportAllResults=TRUE.

`widthPlotClustSM`
: Width of the clusters similarity heatmap in the pdf file. See ?pdf for more details. Default = 7.

`heightPlotClustSM`
: Height of the clusters similarity heatmap in the pdf file. See ?pdf for more details. Default = 5.5.

Details

CONCLUS is a tool for robust clustering and positive marker features selection of single-cell RNA-seq (sc-RNA-seq) datasets. Of note, CONCLUS does not cover the preprocessing steps of sequencing files obtained following next-generation sequencing.

CONCLUS is organized into the following steps:

1) Generation of multiple t-SNE plots with a range of parameters including different selection of genes extracted from PCA.
2) Use the Density-based spatial clustering of applications with noise (DBSCAN) algorithm for idenfication of clusters in each generated t-SNE plot. 3) All DBSCAN results are combined into a cell similarity matrix.
4) The cell similarity matrix is used to define "CONSENSUS" clusters conserved accross the previously defined clustering solutions.
5) Identify marker genes for each concensus cluster. cr

This wrapper function performs the following steps:

1) Building the single-cell RNA-Seq object. See ?scRNAseq-class.
2) Performing the normalization. See ?normaliseCountMatrix.
3) Calculating all tSNEs. See ?generateTSNECoordinates.
4) Clustering with DbScan. See ?runDBSCAN.
5) Computing the cells similarity matrix. See ?clusterCellsInternal.
6) Computing the clusters similarity matrix. If clusToAdd is not NA, add the provided clustering. See ?calculateClustersSimilarity and ?addClustering.
7) Ranking genes. See ?rankGenes.
8) Getting marker genes. See ?retrieveTopClustersMarkers.
9) Getting genes info. See ?retrieveGenesInfo.
10) Plot the cell similarity matrix. See ?plotCellSimilarity.
11) Plot clustered tSNE. See ?plotClusteredTSNE.
12) Plot the cell heatmap. See ?plotCellHeatmap.
13) Plot the clusters similarity heatmap. See ?plotClustersSimilarity.
14) Exporting all results to outputDirectory if exportAllResults=TRUE. See ?exportAllResults.
15) Return an object containing all the results provided by CONCLUS.

If exportAllResults=TRUE, in your "outputDirectory", the sub-folder pictures contains all tSNE with dbscan coloration (sub-folder tSNE_pictures), the cell similarity matrix (Test_cells_correlation_X_clusters.pdf), the cell heatmap (Test_clustersX_meanCenteredTRUE_orderClustersFALSE_orderGenesFALSE markrsPerCluster.pdf‘), and the cluster similarity matrix (‘Test_clusters_similarity_10_clusters.pdf‘). You will also find in the sub-folder ‘Results‘:

+ ‘1_MatrixInfo‘: The normalized count matrix and its meta-data for both rows and columns.
+ ‘2_TSNECoordinates‘: The tSNE coordinates for each parameter of principal components (PCs) and perplexities.

+ '3_dbScan': The different clusters given by DBscan according to different parameters. Each file gives a cluster number for each cell.
+ '4_CellSimilarityMatrix': The matrix underlying the cells similarity heatmap.
+ '5_ClusterSimilarityMatrix': The matrix underlying the clusters similarity heatmap.
+ '6_ConclusResult': A table containing the result of the consensus clustering. This table contains two columns: clusters-cells.
+ '7_fullMarkers': Files containing markers for each cluster, defined by the consensus clustering.
+ '8_TopMarkers': Files containing the top 10 markers for each cluster.
+ '9_genesInfos': Files containing gene information for the top markers defined in the previous folder.

Value

A scRNAseq object containing the similarity matrices and the marker genes.

Author(s)

Nicolas Descostes

Examples

```
experimentName <- "Bergiers"
outputDirectory <- "YourOutputDirectory"
species <- "mouse"

## Load the count matrix
countmatrixPath <- system.file("extdata/countMatrix.tsv", package="conclus")
countMatrix <- loadDataOrMatrix(file=countmatrixPath, type="countMatrix",
                                ignoreCellNumber=TRUE)

## Load the coldata
coldataPath <- system.file("extdata/colData.tsv", package="conclus")
columnsMetaData <- loadDataOrMatrix(file=coldataPath, type="coldata",
columnID="cell_ID")

## Use runCONCLUS
## These parameters are tweaked to fit our example data and reduce
## computing time, please consider using the default parameters or
## adjusted to your dataset.
scr <- runCONCLUS(outputDirectory, experimentName, countMatrix, species,
        columnsMetaData=columnsMetaData, perplexities=c(2,3), tSNENb=1,
        PCs=c(4,5,6,7,8,9,10), epsilon=c(380, 390, 400), minPoints=c(2,3),
        clusterNumber=2)

## Remove the results
unlink(outputDirectory, recursive=TRUE)
```

runDBSCAN,scRNAseq-method
runDBSCAN

Description

Run clustering iterations with selected parameters using DBSCAN.

Usage

```
runDBSCAN(theObject, cores=2, epsilon=c(1.3, 1.4, 1.5), minPoints=c(3, 4),
writeOutput=FALSE)
```

Arguments

theObject	An Object of class scRNASeq for which the count matrix was normalized and the tSNE coordinates were calculated. See ?normaliseCountMatrix and ?generateTSNECoordinates.
cores	Maximum number of jobs that CONCLUS can run in parallel. Default is 1.
epsilon	Reachability distance parameter of fpc::dbscan() function. See Ester et al. (1996) for more details. Default = c(1.3, 1.4, 1.5)
minPoints	Reachability minimum no. of points parameter of fpc::dbscan() function. See Ester et al. (1996) for more details. Default = c(3, 4)
writeOutput	If TRUE, write the results of the dbScan clustering to the output directory defined in theObject, in the sub-directory output_tables. Default = FALSE.

Details

Following the calculation of t-SNE coordinates, DBSCAN is run with a range of epsilon and MinPoints values which will yield a total of 84 clustering solutions (PCs x perplexities x MinPoints x epsilon). minPoints is the minimum cluster size which you assume to be meaningful for your experiment and epsilon is the radius around the cell where the algorithm will try to find minPoints dots. Optimal epsilon must lay one the knee of the k-NN function as shown in the "test_clustering/distance_graph.pdf".

Value

An object of class scRNASeq with its dbscanList slot updated. Also writes the clustering results in "dataDirectory/output_tables" subfolder if the parameter writeOutput is TRUE.

Author(s)

Ilyess RACHEDI, based on code by Polina PAVLOVICH and Nicolas DESCOSTES.

See Also

normaliseCountMatrix generateTSNECoordinates

Examples

```
## Object scr containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Perform the clustering with dbScan
## These parameters are tweaked to fit our example data and reduce
## computing time, please consider using the default parameters or
## adjusted to your dataset.
scr <- runDBSCAN(scr, epsilon=c(380, 390, 400), minPoints=c(2,3), cores=2)
```

`scRNAseq-class` *The scRNAseq class*

Description

S4 class and the main class used by CONCLUS containing the results of the different steps to analyse rare cell populations.

Slots

`experimentName` 'character' string representing the name of the experiment.

`countMatrix` An 'integer matrix' representing the raw count matrix with reads or unique molecular identifiers (UMIs).

`sceNorm` Object of class SingleCellExperiment that contains the colData giving informations about cells and the rowData giving informations about genes. It also contains the normalized count matrix.

`species` 'character' string representing the species of interest. Currently limited to "mouse" and "human". Other organisms can be added on demand.

`outputDirectory` A 'character' string of the path to the root output folder.

`tSNEList` List of 'Tsne' objects representing the different tSNE coordinates generated by CONCLUS.

`dbscanList` List of 'Dbscan' objects representing the different Dbscan clustering generated by CONCLUS.

`suggestedClustersNumber` A number got from the dbscan list representing a suggested clusters number to use in clusterCellsInternal().

`cellsSimilarityMatrix` A numeric Matrix defining how many times two cells have been associated to the same cluster across the 84 solutions (by default) of clustering.

`clustersSimilarityMatrix` A numeric matrix comparing the robustness of the consensus clusters.

`clustersSimiliratyOrdered` A factor representing the clusters ordered by similarity.

`markerGenesList` List of data.frames. Each data frame contains the ranked genes of one cluster.

`topMarkers` A data frame containing the top 10 (by default) marker genes of each clusters.

`genesInfos` A data frame containing informations of the markers genes for each clusters.

Constructor

singlecellRNAseq(experimentName = "character", countMatrix = "matrix", species = "character", outputDirectory = "character")

experimentName: String of the name of the experiment.

countMatrix: Matrix containing the raw counts.

species: 'character' string representing the species of interest. Shoud be mouse or human. Other organisms can be added on demand.

outputDirectory: 'character' string representing the path to the output directory.

Accessors

In the following snippets, x is a scRNAseq object.

getExperimentName(x): Get the name of the experiment.
getCountMatrix(x): Get the count matrix.
getSceNorm(x): Get the SingleCellExperiment object used
getSpecies(x): Get the species.
getOutputDirectory(x): Get the path of the output directory.
getTSNEList(x): Get the list of Tsne objects.
getDbscanList(x): Get the list of Dbscan objects.
getSuggestedClustersNumber(x): Get the suggested clusters number.
getCellsSimilarityMatrix(x): Get the cell similarity matrix.
getClustersSimilarityMatrix(x): Get the cluster similarity matrix.
getClustersSimilarityOrdered(x): Get the clusters ordered by similarity.
getMarkerGenesList(x): Get the list of marker genes by clusters.
getTopMarkers(x): Get the most significant markers by clusters into a data.frame.
getGenesInfos(x): Get a data frame containing informations about marker genes.

Subsetting

In the following snippets, x is a scRNAseq object.

setExperimentName(x): Set the name of the experiment.
setCountMatrix(x): Set the count matrix.
setSceNorm(x): Set the SingleCellExperiment object used.
setSpecies(x): Set the species.
setOutputDirectory(x): Set the path of the output directory.
setTSNEList(x): Set the list of Tsne objects.
setDbscanList(x): Set the list of Dbscan objects.
setCellsSimilarityMatrix(x): Set the cell similarity matrix.
setClustersSimilarityMatrix(x): Set the cluster similarity matrix.
setClustersSimiliratyOrdered(x): Set the clusters ordered by similarity.
setMarkerGenesList(x): Set the list of marker genes by clusters
setTopMarkers(x): Set the most significant markers by clusters.
setGenesInfos(x): Set a data.frame containing informations about the marker genes.

Author(s)

Ilyess Rachedi and Nicolas Descostes

See Also

singlecellRNAseq

---

setters *setters*

Description

Update a slot of a scRNA-seq, Tsne or Dbscan object.

Usage

```
setExperimentName(theObject) <- value

setCountMatrix(theObject) <- value

setSceNorm(theObject) <- value

setSpecies(theObject) <- value

setOutputDirectory(theObject) <- value

setTSNEList(theObject) <- value

setDbscanList(theObject) <- value

setSuggestedClustersNumber(theObject) <- value

setCellsSimilarityMatrix(theObject) <- value

setClustersSimilarityMatrix(theObject) <- value

setClustersSimiliratyOrdered(theObject) <- value

setMarkerGenesList(theObject) <- value

setTopMarkers(theObject) <- value

setGenesInfos(theObject) <- value

## S4 replacement method for signature 'Tsne'
setName(theObject) <- value
```

```
setPC(theObject) <- value

setPerplexity(theObject) <- value

setCoordinates(theObject) <- value

setName(theObject) <- value

setEpsilon(theObject) <- value

setMinPoints(theObject) <- value

setClustering(theObject) <- value
```

Arguments

theObject	A scRNA-seq, Tsne or Dbscan object to update. See description or ?scRNAseq, ?Tsne or ?Dbscan.
value	The value to update the slot with. See ?scRNAseq, ?Tsne or ?Dbscan.

Value

setExperimentName: Update the experiment name slot with a character string (scRNA-seq).

setCountMatrix: Update the countMatrix slot with a matrix of numeric (scRNA-seq).

setSceNorm: Update the normalized countMatrix slot with SingleCellExperiment object (scRNA-seq).

setSpecies: Update the species slot with a character string. Value should be mouse or human. Other organisms can be added on demand (scRNA-seq).

setOutputDirectory: Update the outputDirectory slot with a character string. Value should be a valid path (scRNA-seq).

setTSNEList: Update the tSNEList slot with a list of tSNE objects. See ?Tsne-class (scRNA-seq).

setDbscanList: Update the dbscanList slot with a list of dbscan objects. See ?Dbscan-class (scRNA-seq).

setSuggestedClustersNumber: Update the suggestedClustersNumber slot.

setCellsSimilarityMatrix: Update the cellsSimilarityMatrix slot with a numeric matrix (scRNA-seq).

setClustersSimilarityMatrix: Update the clustersSimilarityMatrix slot with a numeric matrix (scRNA-seq).

setClustersSimiliratyOrdered: Update the clustersSimilarityOrdered slot with a numeric factor (scRNA-seq).

setMarkerGenesList: Update the markerGenesList slot with a list of data frame. The data frame structure should be: data.frame(Gene = c("gene1"), mean_log10_fdr = c(NA), n_05 = c(NA), score = c(NA)) (scRNA-seq).

setTopMarkers: Update the topMarkers slot with a data frame. The data frame structure should be: data.frame(geneName="gene1", clusters=NA). (scRNA-seq)

setGenesInfos: Update the genesInfos slot with a data frame. The data frame structure should be: data.frame(uniprot_gn_symbol=c("symbol"), clusters="1", external_gene_name="gene", go_id="GO1,GO2", mgi_description="description", entrezgene_description="descr", gene_biotype="gene", chromosome_name="1", Symbol="symbol", ensembl_gene_id="ENS", mgi_id="MGI", entrezgene_id="1", uniprot_gn_id="ID"). (scRNA-seq)

setPC: Update the pc slot with a vector of numeric (Tsne).

setPerplexity: Update the perplexity slot with a vector of numeric (Tsne).

setCoordinates: Update the coordinates slot with a matrix of numeric (Tsne).

setName: Update the Tsne or Dbscan name slot with a character string (Dbscan).

setEpsilon: Update the epsilon slot with a vector of numeric (Dbscan).

setMinPoints: Update the minPoints slot with a vector of numeric (Dbscan).

setClustering: Update the clustering slot with a matrix of numeric (Dbscan).

Author(s)

Ilyess RACHEDI

Examples

```
## Load the count matrix
countmatrixPath <- system.file("extdata/countMatrix.tsv", package="conclus")
countMatrix <- loadDataOrMatrix(file=countmatrixPath, type="countMatrix",
                                ignoreCellNumber=TRUE)

## Load the coldata
coldataPath <- system.file("extdata/colData.tsv", package="conclus")
columnsMetaData <- loadDataOrMatrix(file=coldataPath, type="coldata",
                                    columnID="cell_ID")

## Create the initial object
scr <- singlecellRNAseq(experimentName = "Bergiers",
                countMatrix     = countMatrix,
                species         = "mouse",
                outputDirectory = "YourOutputDirectory")

setExperimentName(scr) <- "newName"
setCountMatrix(scr) <- countMatrix[seq_len(10), seq_len(10)]
setSpecies(scr) <- "human"
setOutputDirectory(scr) <- "newPath"
```

`testClustering,scRNAseq-method`
testClustering

Description

This function generates a single clustering iteration of CONCLUS to check whether the chosen parameters of tSNE and dbscan are suitable for your data.

Usage

```
testClustering(theObject, dbscanEpsilon=1.4, minPts=5,
                perplexities=30, PCs=4, randomSeed=42, width=7, height=7,
                cores=2, writeOutput=FALSE, fileTSNE="test_tSNE.pdf",
                fileDist="distance_graph.pdf",
                fileClust="test_clustering.pdf", silent=FALSE, plotKNN=TRUE,
                ...)
```

Arguments

theObject	An Object of class scRNASeq for which the count matrix was normalized. See ?normaliseCountMatrix.
dbscanEpsilon	Single value for the distance parameter of dbscan. Default = 1.4. See ?runDBSCAN for more details.
minPts	Single value for the minimum no. of points parameter of dbscan. Default = 5. See ?runDBSCAN for more details.
perplexities	A single value of perplexity (t-SNE parameter). Default = 30. See ?generateTSNECoordinates for details.
PCs	Single value of first principal components. Default=4. See ?generateTSNECoordinates for details.
randomSeed	Default is 42. Seeds used to generate the tSNE.
width	Width of the pdf file. Default=7. See ?pdf for details.
height	Height of the pdf file. Default=7. See ?pdf for details.
cores	Maximum number of jobs that CONCLUS can run in parallel. Default is 1.
writeOutput	If TRUE, write the results of the test to the output directory defined in theObject in the sub-directory 'test_clustering'. Default = FALSE.
fileTSNE	Name of the pdf file for tSNE. Default="test_tSNE.pdf".
fileDist	Name of the pdf file for NN distance. Default="distance_graph.pdf"
fileClust	Name of the pdf file for dbscan. Default="test_clustering.pdf"
silent	If TRUE, do not plot graphics. Default=FALSE.
plotKNN	If TRUE, output the kNN plot on graphics. Default=TRUE.
...	Options for generating the pdf files. See ?pdf for a list.

Details

The TestClustering function runs one clustering round out of the 84 (default) rounds that CONCLUS normally performs. This step can be useful to determine if the default DBSCAN parameters are suitable for your dataset. By default, they are dbscanEpsilon = c(1.3, 1.4, 1.5) and minPts = c(3,4). If the dashed horizontal line in the k-NN distance plot lays on the "knee" of the curve, it means that

optimal epsilon is equal to the intersection of the line to the y-axis. In our example, optimal epsilon is 1.4 for 5-NN distance where 5 corresponds to MinPts.

In the "test_clustering" folder under outputDirectory, the three plots will be saved where one corresponds to the "distance_graph.pdf", another one to "test_tSNE.pdf", and the last one will be saved as "test_clustering.pdf".

Value

A ggplot object of the tSNE and the dbscan clustering.

Author(s)

Ilyess RACHEDI, based on code by Konstantin CHUKREV and Nicolas DESCOSTES.

See Also

normaliseCountMatrix runDBSCAN pdf

Examples

```
## Object containing the results of previous steps
load(system.file("extdata/scrFull.Rdat", package="conclus"))

## Test the clustering writing pdfs to test_clustering folder
## These parameters are tweaked to fit our example data and reduce
## computing time, please consider using the default parameters or
## adjusted to your dataset.
testClustering(scr, dbscanEpsilon=380, minPts=2, perplexities=2, PCs=4,
                silent=TRUE, writeOutput=TRUE)

## Removing the written results
unlink("YourOutputDirectory/", recursive = TRUE)
```

Tsne-class *The Tsne class*

Description

S4 class containing the features to plot tSNEs. This constructor is internal and is used by the method generateTSNECoordinates.

Details

Tsne is a vector of principal components (PC) and perplexity that are the parameters necessary to reduce the dimensionality of the data in the the form of a t-distributed stochastic neighbor embedding (t-SNE). For details about perplexities parameter see '?Rtsne'.

Slots

`name` A 'character' string representing the name of the tSNE coordinates.

`pc` A 'numeric' value representing the number of principal components used by CONCLUS to perfom a PCA before calculating the tSNE.

`perplexity` A 'numeric' vector. Default: c(30, 40)

`coordinates` A 'numeric' matrix that contains the coordinates of one tSNE solution.

Constructor

Tsne(name = "character", pc = "numeric", perplexity = "numeric", coordinates = "matrix")

name: Empty character string or the name of the tSNE.
pc: Empty 'numeric' number of PCs.
perplexity: Empty 'numeric' perplexity values.
coordinates: Empty 'numeric' "matrix" or matrix of coordinates.

Accessors

In the following snippets, x is a Tsne object.

getName(x): Get the name of the tSNE.
getPC(x): Get the PC used.
getPerplexity(x): Get the perplexity used.
getCoordinates(x): Get the matrix of tSNE coordinates.

Subsetting

In the following snippets, x is a Tsne object.

setName(x) <- value: Set the name of the tSNE.
setPC(x) <- value: Set the PC parameter.
setPerplexity(x) <- value: Set the perplexity parameter.
setCoordinates(x) <- value: Set the matrix of tSNE coordinates.

Author(s)

Ilyess Rachedi and Nicolas Descostes

See Also

generateTSNECoordinates

Index