Package ‘SC3’

December 22, 2016

Type Package
Title Single-Cell Consensus Clustering
Version 1.3.11
Date 2016-12-12
Author Vladimir Kiselev
Maintainer Vladimir Kiselev <vladimir.yu.kiselev@gmail.com>
Description A tool for unsupervised clustering and analysis of single cell RNA-Seq data.
License GPL-3
Imports graphics, stats, utils, methods, e1071, parallel, foreach, doParallel, doRNG, shiny, ggplot2, pheatmap (>= 1.0.8), ROCR, robustbase, rrcov, cluster, WriteXLS, Rcpp (>= 0.11.1), scater
Depends R(>= 3.3)
LinkingTo Rcpp, RcppArmadillo
LazyData TRUE
RoxygenNote 5.0.1
Suggests knitr, rmarkdown, testthat, mclust
VignetteBuilder knitr
biocViews SingleCell, Software, Classification, Clustering, DimensionReduction, SupportVectorMachine, RNASEq, Visualization, Transcriptomics, DataRepresentation, GUI, DifferentialExpression, Transcription
NeedsCompilation no
URL https://github.com/hemberg-lab/SC3
BugReports https://support.bioconductor.org/t/sc3/

R topics documented:
calculate_distance ................................................. 2
calculate_stability ................................................ 3
consensus_matrix .................................................. 3
consmx .............................................................. 4
ED1 ................................................................. 4
ED2 ................................................................. 4
**calculate_distance**

Calculate a distance matrix

**Description**

Distance between the cells, i.e. columns, in the input expression matrix are calculated using the Euclidean, Pearson and Spearman metrics to construct distance matrices.

**Usage**

\[
\text{calculate_distance}(\text{data, method})
\]

**Arguments**

- **data**  expression matrix
- **method**  one of the distance metrics: 'spearman', 'pearson', 'euclidean'
calculate_stability

Value
distance matrix

calculate_stability  Calculate the stability index of the obtained clusters when changing k

Description
Stability index shows how stable each cluster is across the selected range of k. The stability index varies between 0 and 1, where 1 means that the same cluster appears in every solution for different k.

Usage
calculate_stability(consensus, k)

Arguments
consensus  consensus item of the sc3 slot of an object of `SCESet` class
k  number of clusters k

Details
Formula (imagine a given cluster is split into N clusters when k is changed, and in each of the new clusters there are given_cells of the given cluster and also some extra_cells from other clusters): SI = \[ \frac{\sum_{k} \sum_{clusters} \frac{N(given\_cells)}{N(given\_cells + extra\_cells)}}{N\text{(corrects for stability of each cluster)}} / N\text{(corrects for the number of clusters)} / \text{length(ks)} \]

Value
a numeric vector containing a stability index of each cluster

consensus_matrix  Calculate consensus matrix

Description
Consensus matrix is calculated using the Cluster-based Similarity Partitioning Algorithm (CSPA). For each clustering solution a binary similarity matrix is constructed from the corresponding cell labels: if two cells belong to the same cluster, their similarity is 1, otherwise the similarity is 0. A consensus matrix is calculated by averaging all similarity matrices.

Usage
consensus_matrix(clusts)

Arguments
clusts  a matrix containing clustering solutions in columns
Value

consensus matrix

---

**consmx**

*Consensus matrix computation*

**Description**

Computes consensus matrix given cluster labels

**Usage**

```
consmx(dat)
```

**Arguments**

- **dat**
  a matrix containing clustering solutions in columns

---

**ED1**

*Compute Euclidean distance matrix by rows*

**Description**

Used in consmx function

**Usage**

```
ED1(x)
```

**Arguments**

- **x**
  A numeric matrix.

---

**ED2**

*Compute Euclidean distance matrix by columns*

**Description**

Used in sc3-funcs.R distance matrix calculation and within the consensus clustering.

**Usage**

```
ED2(x)
```

**Arguments**

- **x**
  A numeric matrix.
**estkTW**

*Estimate the optimal k for k-means clustering*

**Description**

The function finds the eigenvalues of the sample covariance matrix. It will then return the number of significant eigenvalues according to the Tracy-Widom test.

**Usage**

```r
estkTW(dataset)
```

**Arguments**

- **dataset**
  - processed input expression matrix.

**Value**

- an estimated number of clusters k

---

**get_auroc**

*Calculate the area under the ROC curve for a given gene.*

**Description**

For a given gene a binary classifier is constructed based on the mean cluster expression values (these are calculated using the cell labels). The classifier prediction is then calculated using the gene expression ranks. The area under the receiver operating characteristic (ROC) curve is used to quantify the accuracy of the prediction. A p-value is assigned to each gene by using the Wilcoxon signed rank test.

**Usage**

```r
get_auroc(gene, labels)
```

**Arguments**

- **gene**
  - expression data of a given gene
- **labels**
  - cell labels corresponding to the expression values of the gene
get_de_genes  
Find differentially expressed genes

Description
Differential expression is calculated using the non-parametric Kruskal-Wallis test. A significant p-value indicates that gene expression in at least one cluster stochastically dominates one other cluster. Note that the calculation of differential expression after clustering can introduce a bias in the distribution of p-values, and thus we advise to use the p-values for ranking the genes only.

Usage
get_de_genes(dataset, labels)

Arguments
- dataset: expression matrix
- labels: cell labels corresponding to the columns of the expression matrix

Value
a numeric vector containing the differentially expressed genes and corresponding p-values

Examples
d <- get_de_genes(treutlein[1:10, ], colnames(treutlein))
head(d)

get_marker_genes  
Calculate marker genes

Description
Find marker genes in the dataset. The get_auroc is used to calculate marker values for each gene.

Usage
get_marker_genes(dataset, labels)

Arguments
- dataset: expression matrix
- labels: cell labels corresponding clusters

Value
data.frame containing the marker genes, corresponding cluster indexes and adjusted p-values
**get_outl_cells**

**Examples**

```r
d <- get_marker_genes(treutlein[1:10,], colnames(treutlein))
d
```

---

**Description**

Outlier cells in each cluster are detected using robust distances, calculated using the minimum covariance determinant (MCD), namely using `covMcd`. The outlier score shows how different a cell is from all other cells in the cluster and it is defined as the differences between the square root of the robust distance and the square root of the 99.99

**Usage**

```r
get_outl_cells(dataset, labels)
```

**Arguments**

- `dataset`: expression matrix
- `labels`: cell labels corresponding to the columns of the expression matrix

**Value**

a numeric vector containing the cell labels and corresponding outlier scores ordered by the labels

**Examples**

```r
d <- get_outl_cells(treutlein[1:10,], colnames(treutlein))
head(d)
```

---

**get_processed_dataset**

*Get processed dataset used by SC3 from the default scater slots*

**Description**

Takes data from the `exprs_values` slot, applies gene filter and log transformation.

**Usage**

```r
get_processed_dataset(object)
```

**Arguments**

- `object`: an object of `SCESet` class
organise_de_genes

markers_for_heatmap  
Reorder and subset gene markers for plotting on a heatmap

Description
Reorders the rows of the input data.frame based on the sc3_k_markers_clusts column and also keeps only the top 10 genes for each value of sc3_k_markers_clusts.

Usage
markers_for_heatmap(markers)

Arguments

markers  
a data.frame object with the following colnames: sc3_k_markers_clusts, sc3_k_markers_auroc, sc3_k_markers_padj.

norm_laplacian  
Graph Laplacian calculation

Description
Calculate graph Laplacian of a symmetrix matrix

Usage
norm_laplacian(A)

Arguments

A  
symmetric matrix

organise_de_genes  
Get differentially expressed genes from an object of SCESet class

Description
This functions returns all marker gene columns from the phenoData slot of the input object corresponding to the number of clusters k. Additionally, it rearranges genes by the cluster index and order them by the area under the ROC curve value inside of each cluster.

Usage
organise_de_genes(object, k, p_val)

Arguments

object  
an objec of SCESet class
k  
number of cluster
p_val  
p-value threshold
organise_marker_genes  Get marker genes from an object of SCESet class

Description

This function returns all marker gene columns from the phenoData slot of the input object corresponding to the number of clusters k. Additionally, it rearranges genes by the cluster index and order them by the area under the ROC curve value inside of each cluster.

Usage

organise_marker_genes(object, k, p_val, auroc)

Arguments

object  an object of SCESet class
k  number of cluster
p_val  p-value threshold
auroc  area under the ROC curve threshold

prepare_for_svm  A helper function for the SVM analysis

Description

Defines train and study cell indeces based on the svm_num_cells and svm_train_inds input parameters

Usage

prepare_for_svm(N, svm_num_cells = NULL, svm_train_inds = NULL, svm_max)

Arguments

N  number of cells in the input dataset
svm_num_cells  number of random cells to be used for training
svm_train_inds  indeces of cells to be used for training
svm_max  define the maximum number of cells below which SVM is not run

Value

A list of indeces of the train and the study cells
reindex_clusters  

Reindex cluster labels in ascending order

Description

Given an hclust object and the number of clusters k this function reindex the clusters inferred by cutree(hc, k)[hc$order], so that they appear in ascending order. This is particularly useful when plotting heatmaps in which the clusters should be numbered from left to right.

Usage

reindex_clusters(hc, k)

Arguments

hc  
an object of class hclust

k  
number of cluster to be inferred from hc

Examples

hc <- hclust(dist(USArrests), 'ave')
cutree(hc, 10)[hc$order]
reindex_clusters(hc, 10)[hc$order]

sc3  

Run all steps of SC3 in one go

Description

This function is a wrapper that executes all steps of SC3 analysis in one go.

Usage

sc3.SCESet(object, ks = NULL, exprs_values = "exprs", gene_filter = TRUE, pct_dropout_min = 10, pct_dropout_max = 90, d_region_min = 0.04, d_region_max = 0.07, svm_num_cells = NULL, svm_train_inds = NULL, svm_max = 5000, n_cores = NULL, kmeans_nstart = NULL, kmeans_iter_max = 1e+09, k_estimator = FALSE, biology = FALSE, rand_seed = 1)

## S4 method for signature 'SCESet'
sc3(object, ks = NULL, exprs_values = "exprs", gene_filter = TRUE, pct_dropout_min = 10, pct_dropout_max = 90, d_region_min = 0.04, d_region_max = 0.07, svm_num_cells = NULL, svm_train_inds = NULL, svm_max = 5000, n_cores = NULL, kmeans_nstart = NULL, kmeans_iter_max = 1e+09, k_estimator = FALSE, biology = FALSE, rand_seed = 1)
Arguments

**object**  
an object of SCESet class.

**ks**  
a range of the number of clusters k used for SC3 clustering. Can also be a single integer.

**exprs_values**  
character string indicating which values should be used as the expression values for SC3 clustering. Valid value is any named element of the assayData slot of the SCESet object. Default is 'exprs'. See `get_exprs` function of the scater package for more details.

**gene_filter**  
a boolean variable which defines whether to perform gene filtering before SC3 clustering.

**pct_dropout_min**  
if `gene_filter` = TRUE, then genes with percent of dropouts smaller than `pct_dropout_min` are filtered out before clustering.

**pct_dropout_max**  
if `gene_filter` = TRUE, then genes with percent of dropouts larger than `pct_dropout_max` are filtered out before clustering.

**d_region_min**  
defines the minimum number of eigenvectors used for kmeans clustering as a fraction of the total number of cells. Default is 0.04. See SC3 paper for more details.

**d_region_max**  
defines the maximum number of eigenvectors used for kmeans clustering as a fraction of the total number of cells. Default is 0.07. See SC3 paper for more details.

**svm_num_cells**  	number of randomly selected training cells to be used for SVM prediction. The default is NULL.

**svm_train_inds**  
a numeric vector defining indeces of training cells that should be used for SVM training. The default is NULL.

**svm_max**  
define the maximum number of cells below which SVM is not run.

**n_cores**  
defines the number of cores to be used on the user’s machine.

**kmeans_nstart**  
nstart parameter passed to `kmeans` function. Can be set manually. By default it is 1000 for up to 2000 cells and 50 for more than 2000 cells.

**kmeans_iter_max**  
iter.max parameter passed to `kmeans` function.

**k_estimator**  
boolean parameter, defines whether to estimate an optimal number of clusters k.

**biology**  
boolean parameter, defines whether to compute differentially expressed genes, marker genes and cell outliers.

**rand_seed**  
sets the seed of the random number generator. SC3 is a stochastic method, so setting the rand_seed to a fixed values can be used for reproducibility purposes.

**...**  
further arguments passed to `sc3.SCESet`

Value

an object of SCESet class
sc3_calc_biology

Calculate DE genes, marker genes and cell outliers.

Description

This function calculates differentially expressed (DE) genes, marker genes and cell outliers based on the consensus SC3 clusterings.

Usage

sc3_calc_biology.SCESet(object)

## S4 method for signature 'SCESet'
sc3_calc_biology(object)

Arguments

object an object of 'SCESet' class

Details

DE genes are calculated using `get_de_genes`. Results of the DE analysis are saved as new columns in the `featureData` slot of the input object. The column names correspond to the adjusted p-values of the genes and have the following format: `sc3_k_de_padj`, where k is the number of clusters.

Marker genes are calculated using `get_marker_genes`. Results of the marker gene analysis are saved as three new columns (for each k) to the `featureData` slot of the input object. The column names correspond to the SC3 cluster labels, to the adjusted p-values of the genes and to the area under the ROC curve and have the following format: `sc3_k_markers_clusts`, `sc3_k_markers_padj` and `sc3_k_markers_auroc`, where k is the number of clusters.

Outlier cells are calculated using `get_outl_cells`. Results of the cell outlier analysis are saved as new columns in the `phenoData` slot of the input object. The column names correspond to the log2(outlier_score) and have the following format: `sc3_k_log2_outlier_score`, where k is the number of clusters.

Additionally, `biology` item is added to the `sc3` slot and is set to TRUE indicating that the biological analysis of the dataset has been performed.

Value

an object of 'SCESet' class
**sc3_calc_consens**

*Calculate consensus matrix.*

**Description**

This function calculates consensus matrices based on the clustering solutions contained in the kmeans item of the sc3 slot of the SCESet object. It then creates and populates the consensus item of the sc3 slot with consensus matrices, their hierarchical clusterings in hclust objects, and Silhouette indeces of the clusters. It also removes the previously calculated kmeans clusterings from the sc3 slot, as they are not needed for further analysis.

**Usage**

```r
sc3_calc_consens.SCESet(object)
```

## S4 method for signature 'SCESet'

```r
sc3_calc_consens(object)
```

**Arguments**

- `object` an object of `SCESet` class

**Details**

Additionally, it also adds new columns to the phenoData slot of the input object. The column names correspond to the consensus cell labels and have the following format: `sc3_k_clusters`, where `k` is the number of clusters.

**Value**

an object of `SCESet` class

---

**sc3_calc_dists**

*Calculate distances between the cells.*

**Description**

This function calculates distances between the cells contained in the processed_dataset item of the sc3 slot of the SCESet object. It then creates and populates the following items of the sc3 slot:

- distances - contains a list of distance matrices corresponding to Euclidean, Pearson and Spearman distances.

**Usage**

```r
sc3_calc_dists.SCESet(object)
```

## S4 method for signature 'SCESet'

```r
sc3_calc_dists(object)
```
sc3_estimate_k

Arguments
object an object of 'SCESet' class

Value
an object of 'SCESet' class

sc3_calc_transfs Calculate transformations of the distance matrices.

Description
This function transforms all distances items of the sc3 slot of the SCESet object using either principal component analysis (PCA) or by calculating the eigenvectors of the associated graph Laplacian. The columns of the resulting matrices are then sorted in descending order by their corresponding eigenvalues. The first d columns (where \( d = \max(object@sc3$n\_dim) \)) of each transformation are then written to the transformations item of the sc3 slot. Additionally, this function also removes the previously calculated distances from the sc3 slot, as they are not needed for further analysis.

Usage
sc3_calc_transfs.SCESet(object)

## S4 method for signature 'SCESet'
s3_calc_transfs(object)

Arguments
object an object of 'SCESet' class

Value
an object of 'SCESet' class

sc3_estimate_k Estimate the optimal k for k-means clustering

Description
Uses Tracy-Widom theory on random matrices to estimate the optimal number of clusters \( k \). Using the function estkTW to perform the estimation. It creates and populates the following items of the 'sc3' slot:

- k_estimation - contains the estimated value of \( k \).
**Usage**

```r
sc3_estimate_k.SCESet(object)
```

## S4 method for signature 'SCESet'

```r
sc3_estimate_k(object)
```

**Arguments**

- `object`  
an object of SCESet class

**Value**

- an estimated value of k

---

### sc3_export_results_xls

**Write SC3 results to Excel file**

**Description**

This function writes all SC3 results to an excel file.

**Usage**

```r
sc3_export_results_xls.SCESet(object, filename = "sc3_results.xls")
```

## S4 method for signature 'SCESet'

```r
sc3_export_results_xls(object,
    filename = "sc3_results.xls")
```

**Arguments**

- `object`  
an object of `SCESet` class
- `filename`  
name of the excel file, to which the results will be written

---

### sc3_interactive

**Opens SC3 results in an interactive session in a web browser.**

**Description**

Runs interactive shiny session of SC3 based on precomputed clusterings.

**Usage**

```r
sc3_interactive.SCESet(object)
```

## S4 method for signature 'SCESet'

```r
sc3_interactive(object)
```
sc3_kmeans

Arguments

object an object of SCESet class

Value

Opens a browser window with an interactive shiny app and visualize all precomputed clusterings.

sc3_kmeans kmeans clustering of cells.

Description

This function performs kmeans clustering of the matrices contained in the transformations item of the sc3 slot of the SCESet object. It then creates and populates the following items of the sc3 slot:

- kmeans - contains a list of kmeans clusterings.

Additionally, it also removes the previously calculated transformations from the sc3 slot, as they are not needed for further analysis.

Usage

sc3_kmeans.SCESet(object)

## S4 method for signature 'SCESet'
sc3_kmeans(object)

Arguments

object an object of ’SCESet’ class

Details

See sc3_prepare for the default clustering parameters.

Value

an object of ’SCESet’ class
sc3_plot_cluster_stability

*Plot stability of the clusters*

**Description**
Stability index shows how stable each cluster is across the selected range of ks. The stability index varies between 0 and 1, where 1 means that the same cluster appears in every solution for different k.

**Usage**

```
sc3_plot_cluster_stability.SCESet(object, k)
```

### S4 method for signature 'SCESet'
```
sc3_plot_cluster_stability(object, k)
```

**Arguments**
- **object**: an object of `SCESet` class
- **k**: number of clusters

sc3_plot_consensus

*Plot consensus matrix as a heatmap*

**Description**

The consensus matrix is a NxN matrix, where N is the number of cells. It represents similarity between the cells based on the averaging of clustering results from all combinations of clustering parameters. Similarity 0 (blue) means that the two cells are always assigned to different clusters. In contrast, similarity 1 (red) means that the two cells are always assigned to the same cluster. The consensus matrix is clustered by hierarchical clustering and has a diagonal-block structure. Intuitively, the perfect clustering is achieved when all diagonal blocks are completely red and all off-diagonal elements are completely blue.

**Usage**

```
sc3_plot_consensus.SCESet(object, k, show_pdata = NULL)
```

### S4 method for signature 'SCESet'
```
sc3_plot_consensus(object, k, show_pdata = NULL)
```

**Arguments**
- **object**: an object of `SCESet` class
- **k**: number of clusters
- **show_pdata**: a vector of colnames of the pData(object) table. Default is NULL. If not NULL will add pData annotations to the columns of the output matrix
sc3_plot_de_genes  
**Plot expression of DE genes of the clusters identified by SC3 as a heatmap**

**Description**
SC3 plots gene expression profiles of the 50 genes with the lowest p-values.

**Usage**
```r
sc3_plot_de_genes.SCESet(object, k, p.val = 0.01, show_pdata = NULL)
```

## S4 method for signature 'SCESet'
sc3_plot_de_genes(object, k, p.val = 0.01, 
show_pdata = NULL)

**Arguments**
- **object**: an object of 'SCESet' class
- **k**: number of clusters
- **p.val**: significance threshold used for the DE genes
- **show_pdata**: a vector of colnames of the pData(object) table. Default is NULL. If not NULL will add pData annotations to the columns of the output matrix

---

sc3_plot_expression  
**Plot expression matrix used for SC3 clustering as a heatmap**

**Description**
The expression panel represents the original input expression matrix (cells in columns and genes in rows) after the gene filter. Genes are clustered by kmeans with k = 100 (dendrogram on the left) and the heatmap represents the expression levels of the gene cluster centers after log2-scaling.

**Usage**
```r
sc3_plot_expression.SCESet(object, k, show_pdata = NULL)
```

## S4 method for signature 'SCESet'
scc3_plot_expression(object, k, show_pdata = NULL)

**Arguments**
- **object**: an object of 'SCESet' class
- **k**: number of clusters
- **show_pdata**: a vector of colnames of the pData(object) table. Default is NULL. If not NULL will add pData annotations to the columns of the output matrix
sc3_plot_markers

Plot expression of marker genes identified by SC3 as a heatmap.

Description
By default the genes with the area under the ROC curve (AUROC) > 0.85 and with the p-value < 0.01 are selected and the top 10 marker genes of each cluster are visualized in this heatmap.

Usage
sc3_plot_markers.SCESet(object, k, auroc = 0.85, p.val = 0.01, show_pdata = NULL)

## S4 method for signature 'SCESet'
sc3_plot_markers(object, k, auroc = 0.85, p.val = 0.01, show_pdata = NULL)

Arguments
- object: an object of 'SCESet' class
- k: number of clusters
- auroc: area under the ROC curve
- p.val: significance threshold used for the DE genes
- show_pdata: a vector of colnames of the pData(object) table. Default is NULL. If not NULL will add pData annotations to the columns of the output matrix

sc3_plot_silhouette

Plot silhouette indexes of the cells

Description
A silhouette is a quantitative measure of the diagonality of the consensus matrix. An average silhouette width (shown at the bottom left of the silhouette plot) varies from 0 to 1, where 1 represents a perfectly block-diagonal consensus matrix and 0 represents a situation where there is no block-diagonal structure. The best clustering is achieved when the average silhouette width is close to 1.

Usage
sc3_plot_silhouette.SCESet(object, k)

## S4 method for signature 'SCESet'
sc3_plot_silhouette(object, k)

Arguments
- object: an object of 'SCESet' class
- k: number of clusters
sc3_prepare

Prepare the SCESet object for SC3 clustering.

Description

This function prepares an object of SCESet class for SC3 clustering. It creates and populates the following items of the sc3 slot of the SCESet object:

- **exprs_values** - the same as the `exprs_values` argument.
- **logged** - a boolean variable which defines whether expression values have been log-transformed. If `exprs_values` != 'exprs' or `object@logged == FALSE` then it is set to TRUE. Otherwise it is set to TRUE. Works correctly for all default elements of the assayData slot of the SCESet object. If during the analysis you create your own element of the assayData slot, please set the logged parameter of the sc3 slot manually and accordingly after running `sc3_prepare`.
- **kmeans_iter_max** - the same as the `kmeans_iter_max` argument.
- **kmeans_nstart** - the same as the `kmeans_nstart` argument.
- **n_dim** - contains numbers of the number of eigenvectors to be used in kmeans clustering.
- **rand_seed** - the same as the `rand_seed` argument.
- **svm_train_inds** - if SVM is used this item contains indexes of the training cells to be used for SC3 clustering and further SVM prediction.
- **svm_study_inds** - if SVM is used this item contains indexes of the cells to be predicted by SVM.
- **n_cores** - the same as the `n_cores` argument.
- **ks** - the same as the `ks` argument.

Usage

```r
sc3_prepare.SCESet(object, exprs_values = "exprs", ks = NULL,
gene_filter = TRUE, pct_dropout_min = 10, pct_dropout_max = 90,
d_region_min = 0.04, d_region_max = 0.07, svm_num_cells = NULL,
svm_train_inds = NULL, svm_max = 5000, n_cores = NULL,
kmeans_nstart = NULL, kmeans_iter_max = 1e+09, rand_seed = 1)
```

```r
## S4 method for signature 'SCESet'
sc3_prepare(object, exprs_values = "exprs", ks = NULL,
gene_filter = TRUE, pct_dropout_min = 10, pct_dropout_max = 90,
d_region_min = 0.04, d_region_max = 0.07, svm_num_cells = NULL,
svm_train_inds = NULL, svm_max = 5000, n_cores = NULL,
kmeans_nstart = NULL, kmeans_iter_max = 1e+09, rand_seed = 1)
```

Arguments

- **object** an object of SCESet class.
- **exprs_values** character string indicating which values should be used as the expression values for SC3 clustering. Valid value is any named element of the assayData slot of the SCESet object. Default is 'exprs'. See `get_exprs` function of the scater package for more details.
### ks
A continuous range of integers - the number of clusters k used for SC3 clustering. Can also be a single integer.

### gene_filter
A boolean variable which defines whether to perform gene filtering before SC3 clustering.

### pct_dropout_min
If `gene_filter = TRUE`, then genes with percent of dropouts smaller than `pct_dropout_min` are filtered out before clustering.

### pct_dropout_max
If `gene_filter = TRUE`, then genes with percent of dropouts larger than `pct_dropout_max` are filtered out before clustering.

### d_region_min
Defines the minimum number of eigenvectors used for kmeans clustering as a fraction of the total number of cells. Default is 0.04. See SC3 paper for more details.

### d_region_max
Defines the maximum number of eigenvectors used for kmeans clustering as a fraction of the total number of cells. Default is 0.07. See SC3 paper for more details.

### svm_num_cells
Number of randomly selected training cells to be used for SVM prediction. The default is NULL.

### svm_train_inds
A numeric vector defining indexes of training cells that should be used for SVM training. The default is NULL.

### svm_max
Define the maximum number of cells below which SVM is not run.

### n_cores
Defines the number of cores to be used on the user’s machine.

### kmeans_nstart
Nstart parameter passed to `kmeans` function. Default is 1000 for up to 2000 cells and 50 for more than 2000 cells.

### kmeans_iter_max
Iter.max parameter passed to `kmeans` function. Default is 1e+09.

### rand_seed
Sets the seed of the random number generator. SC3 is a stochastic method, so setting the `rand_seed` to a fixed value can be used for reproducibility purposes.

### ... Further arguments passed to `sc3_prepare.SCESet`

**Value**

An object of 'SCESet' class

---

**Description**

This method parallelize SVM prediction for each k (the number of clusters). Namely, for each k, `support_vector_machines` function is utilized to predict the labels of study cells. Training cells are selected using `svm_train_inds` item of the sc3 slot of the input SCESet object.

**Usage**

```r
sc3_run_svm.SCESet(object)
```

### S4 method for signature 'SCESet'

```r
sc3_run_svm(object)
```
Arguments

object an object of 'SCESet' class

Details

Results are written to the sc3_k_clusters columns to the phenoData slot of the input object, where k is the number of clusters.

Value

an object of 'SCESet' class

support_vector_machines

Run support vector machines (SVM) prediction

Description

Train an SVM classifier on a training dataset (train) and then classify a study dataset (study) using the classifier.

Usage

support_vector_machines(train, study, kern)

Arguments

train training dataset with colnames, corresponding to training labels
study study dataset
kern kernel to be used with SVM

Value

classification of the study dataset

tmult

Matrix left-multiplied by its transpose

Description

Given matrix A, the procedure returns A'A.

Usage

tmult(x)

Arguments

x Numeric matrix.
**transformation**

**Distance matrix transformation**

**Description**

All distance matrices are transformed using either principal component analysis (PCA) or by calculating the eigenvectors of the graph Laplacian (Spectral). The columns of the resulting matrices are then sorted in descending order by their corresponding eigenvalues.

**Usage**

transformation(dists, method)

**Arguments**

- dists: distance matrix
- method: transformation method: either 'pca' or 'laplacian'

**Value**

transformed distance matrix

**treutlein**

*Single cell RNA-Seq data extracted from a publication by Treutlein et al.*

**Description**

Single cell RNA-Seq data extracted from a publication by Treutlein et al.

**Usage**

treutlein

**Format**

An object of class matrix with 23271 rows and 80 columns.

**Source**


Columns represent cells, rows represent genes expression values. Colnames represent indexes of cell clusters (known information based on the experimental protocol). There are 80 cells and 5 clusters in this dataset.
Index

- Topic **datasets**
  - treutlein, 23
- calculate_distance, 2
- calculate_stability, 3
- consensus_matrix, 3
- consmx, 4
- covMcd, 7
- ED1, 4
- ED2, 4
- estkTW, 5, 14
- get_auroc, 5, 6
- get_de_genes, 6, 12
- get_exprs, 11, 20
- get_marker_genes, 6, 12
- get_outl_cells, 7, 12
- get_processed_dataset, 7
- hclust, 10
- kmeans, 11, 16, 20, 21
- markers_for_heatmap, 8
- norm_laplacian, 8
- organise_de_genes, 8
- organise_marker_genes, 9
- prepare_for_svm, 9
- reindex_clusters, 10
- sc3, 10
- sc3, SCESet-method (sc3), 10
- sc3.SCESet, 11
- sc3.SCESet (sc3), 10
- sc3_calc_biology, 12
- sc3_calc_biology, (sc3_calc_biology), 12
- sc3_calc_biology, SCESet-method (sc3_calc_biology), 12
- sc3_calc_biology, SCESet (sc3_calc_biology), 12
- sc3_calc_consens, 13
- sc3_calc_consens, (sc3_calc_consens), 13
- sc3_calc_consens, SCESet-method (sc3_calc_consens), 13
- sc3_calc_consens, SCESet (sc3_calc_consens), 13
- sc3_calc_dist, 13
- sc3_calc_dist, (sc3_calc_dist), 13
- sc3_calc_dist, SCESet-method (sc3_calc_dist), 13
- sc3_calc_dist, SCESet (sc3_calc_dist), 13
- sc3_calc_dist, SCESet-method (sc3_calc_dist), 13
- sc3_calc_dist, SCESet (sc3_calc_dist), 13
- sc3_calc_transf, 14
- sc3_calc_transf, (sc3_calc_transf), 14
- sc3_calc_transf, SCESet-method (sc3_calc_transf), 14
- sc3_calc_transf, SCESet (sc3_calc_transf), 14
- sc3_estimate_k, 14
- sc3_estimate_k, SCESet-method (sc3_estimate_k), 14
- sc3_estimate_k, SCESet (sc3_estimate_k), 14
- sc3_export_results_xls, 15
- sc3_export_results_xls, (sc3_export_results_xls), 15
- sc3_export_results_xls, SCESet-method (sc3_export_results_xls), 15
- sc3_export_results_xls, SCESet (sc3_export_results_xls), 15
- sc3_interactive, 15
- sc3_interactive, (sc3_interactive), 15
- sc3_interactive, SCESet-method (sc3_interactive), 15
- sc3_interactive, SCESet (sc3_interactive), 15
- sc3_kmeans, 16
- sc3_kmeans, (sc3_kmeans), 16
- sc3_kmeans, SCESet-method (sc3_kmeans), 16
- sc3_kmeans, SCESet (sc3_kmeans), 16
- sc3_plot_cluster_stability, 17
- sc3_plot_cluster_stability, (sc3_plot_cluster_stability), 17
INDEX

17
sc3_plot_cluster_stability, SCESet-method
(sc3_plot_cluster_stability), 17

17
sc3_plot_consensus, 17
(sc3_plot_consensus), 17

17
sc3_plot_expression, 18
(sc3_plot_expression), 18

18
sc3_plot_de_genes, 18
(sc3_plot_de_genes), 18

18
sc3_plot_markers, 19
(sc3_plot_markers), 19

19
sc3_plot_silhouette, 19
(sc3_plot_silhouette), 19

20
sc3_prepare, 16
(sc3_prepare), 20

20
sc3_prepare.SCESet, 17

21
sc3_run_svm, 21
(sc3_run_svm), 21

21
sc3_run_svm.SCESet, 21

21
support_vector_machines, 21, 22

22
tmult, 22
transformation, 23
treutlein, 23