Package ‘SC3’

November 21, 2016

Type Package
Title Single-Cell Consensus Clustering
Version 1.3.6
Date 2016-10-31
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, RSelenium, e1071, parallel, foreach, doParallel, doRNG, shiny, ggplot2, pheatmap (>= 1.0.8), colorspace, ROCR, robustbase, rrcov, cluster, WriteXLS, Rtsne, Rcpp (>= 0.11.1), scater
Depends R (>= 3.3)
LinkingTo Rcpp, RcppArmadillo
LazyData TRUE
RoxygenNote 5.0.1
Suggests knitr, rmarkdown, testthat
VignetteBuilder knitr
biocViews Classification, Clustering, DimensionReduction, SupportVectorMachine, RNASeq, Visualization, Transcriptomics, DataRepresentation, GUI, DifferentialExpression, GeneSetEnrichment, Transcription
NeedsCompilation no
URL https://github.com/hemberg-lab/SC3

R topics documented:

  calculate_distance ........................................... 2
  consensus_matrix ............................................ 3
  consmx ......................................................... 3
  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**

```
calculate_distance(data, method)
```
**consensus_matrix**

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

**Value**
- distance 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
   estkTW(dataset)

Arguments
   dataset  processed input expression matrix.

Value
   an estimated number of clusters k
### Description
The gene filter removes genes/transcripts that are either expressed (expression value is more than 2) in less than X (rare genes/transcripts) or expressed (expression value is more than 0) in at least (100-X)

### Usage
```
gene_filter(data, fraction = 0.06, reads.rare = 2, reads.ubiq = 0)
```

### Arguments
- **data**: input expression matrix
- **fraction**: fraction of cells (X/100).
- **reads.rare**: expression value threshold for genes that are expressed in less than fraction*N cells (rare genes)
- **reads.ubiq**: expression value threshold for genes that are expressed in more than (1-fraction)*N cells (ubiquitous genes)

### Value
a boolean vector representing the gene filter

### Description
If the cell labels are available this functions allows a user to calculate differentially expressed genes manually.

### 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, colnames(treutlein))
head(d)
```
**get_marker_genes**

*Find marker genes*

**Description**

If the cell labels are available this functions allows a user to calculate marker genes manually.

**Usage**

```r
get_marker_genes(dataset, labels)
```

**Arguments**

- `dataset`: expression matrix
- `labels`: cell labels corresponding to the columns of the expression matrix. Labels must be integers or character integers, e.g. 1, 2, 3 or '1', '2', '3' etc.

**Value**

data.frame containing the marker genes

**Examples**

```r
d <- get_marker_genes(treutlein, colnames(treutlein))
head(d)
```

---

**get_outl_cells**

*Find cell outliers*

**Description**

If the cell labels are available this functions allows a user to calculate cell outlier scores manually.

**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, 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**  
get_processed_dataset(object)

**Arguments**  
- **object** an object of `SCESet` class

---

**iwanthue**  
Generate a colour palette by k-means clustering of LAB colour space.

**Description**  
Generate a palette of distinct colours through k-means clustering of LAB colour space.

**Usage**  
iwanthue(n, hmin = 0, hmax = 360, cmin = 0, cmax = 180, lmin = 0, lmax = 100, plot = FALSE, random = FALSE)

**Arguments**  
- **n** Numeric. The number of colours to generate.
- **hmin** Numeric, in the range [0, 360]. The lower limit of the hue range to be clustered.
- **hmax** Numeric, in the range [0, 360]. The upper limit of the hue range to be clustered.
- **cmin** Numeric, in the range [0, 180]. The lower limit of the chroma range to be clustered.
- **cmax** Numeric, in the range [0, 180]. The upper limit of the chroma range to be clustered.
- **lmin** Numeric, in the range [0, 100]. The lower limit of the luminance range to be clustered.
- **lmax** Numeric, in the range [0, 100]. The upper limit of the luminance range to be clustered.
- **plot** Logical. Should the colour swatches be plotted (using `swatch`)?
- **random** Logical. If TRUE, clustering will be determined by the existing RNG state. If FALSE, the seed will be set to 1 for clustering, and on exit, the function will restore the pre-existing RNG state.
Details

Note that iwanthue currently doesn’t support $h_{\text{min}}$ greater than $h_{\text{max}}$ (which should be allowed, since hue is circular).

Value

A vector of $n$ colours (as hexadecimal strings), representing centers of clusters determined through k-means clustering of the LAB colour space delimited by $h_{\text{min}}, h_{\text{max}}, c_{\text{min}}, c_{\text{max}}, l_{\text{min}}$ and $l_{\text{max}}$.

References

- R implementation of iwanthue by John Baumgartner
- iwanthue - colors for data scientists
- iwanthue on GitHub

See Also

swatch

summary

norm_laplacian

Graph Laplacian calculation

Description

Calculate graph Laplacian of a symmetric matrix

Usage

norm_laplacian(A)

Arguments

A symmetric matrix

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)
sc3.SCESet

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

---

**sc3.SCESet**  
*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

```r
sc3.SCESet(object, exprs_values = "exprs", gene.filter = FALSE, 
    gene.filter.fraction = 0.06, gene.reads.rare = 2, gene.reads.ubiq = 0, 
    log.scale = FALSE, d.region.min = 0.04, d.region.max = 0.07, 
    svm.num.cells = NULL, svm.train.inds = NULL, svm.max = 5000, 
    n.cores = NULL, ks = NULL, k.means.nstart = NULL, 
    k.means.iter.max = 1e+09, biology = TRUE, seed = 1)
```

```
## S4 method for signature 'SCESet'
sc3(object, exprs_values = "exprs", gene.filter = FALSE, 
    gene.filter.fraction = 0.06, gene.reads.rare = 2, gene.reads.ubiq = 0, 
    log.scale = FALSE, d.region.min = 0.04, d.region.max = 0.07, 
    svm.num.cells = NULL, svm.train.inds = NULL, svm.max = 5000, 
    n.cores = NULL, ks = NULL, k.means.nstart = NULL, 
    k.means.iter.max = 1e+09, biology = TRUE, 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 arguments are 'tpm' (default; transcripts per million), 'norm_tpm' (normalised TPM values), 'fpkm' (FPKM values), 'norm_fpkm' (normalised FPKM values), 'counts' (counts for each feature), 'norm_counts', 'cpm' (counts-per-million), 'norm_cpm' (normalised counts-per-million), 'exprs' (whatever is in the 'exprs' slot of the SCESet object; default), 'norm_exprs' (normalised expression values) or 'stand_exprs' (standardised expression values) or any other named element of the assayData slot of the SCESet object that can be accessed with the get_exprs function.
gene.filter  
a boolean variable which defines whether to perform gene filtering before SC3  
clustering. Default is TRUE. The gene filter removes genes/transcripts that are  
either expressed (expression value is more than gene.reads.rare) in less than X  
(expression value is more than gene.reads.ubiq) in at least (100*X) cells (ubiqui-
tous genes/transcripts), where X is the gene.filter.fraction*100. The motivation  
for the gene filter is that ubiquitous and rare genes most often are not informative  
for the clustering.

gene.filter.fraction  
fraction of cells. Default is 0.06.

gene.reads.rare  
expression value threshold for rare genes. Default is 2.

gene.reads.ubiq  
expression value threshold for ubiquitous genes. Default is 0.

log.scale  
a boolean variable which defines whether to perform log2 scaling before SC3  
clustering. Default is TRUE.

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.

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.

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.

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

k.means.nstart  
nstart parameter used by kmeans() function. Default is 1000 for up to 2000 cells  
and 50 for more than 2000 cells.

k.means.iter.max  
iter.max parameter passed to kmeans function. Default is 1e+09.

biology  
boolean variable, defines whether to comput DE genes, marker genes and cell  
outliers

seed  
sets seed for the random number generator. Can be used to check the stability  
of clustering results: if the results are the same after changing the seed several  
time, then the clustering solution is stable.

Value

an object of 'SCESet' class
sc3_calc_biology.SCESet

Calculate DE genes, marker genes and cell outliers.

Description
This function calculates DE genes, marker genes and cell outliers based on the consensus clusterings contained in the consensus item of the object@sc3 slot. It then creates and populates the following items of the object@sc3 slot:

• biology - contains lists of DE genes, marker genes and cell outliers data frames.

Usage
sc3_calc_biology.SCESet(object)

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

Arguments
object an object of 'SCESet' class

Value
an object of 'SCESet' class

sc3_calc_consens.SCESet

Calculate consensus matrix.

Description
This function calculates consensus matrices based on the clustering solutions contained in the sc3_kmeans item of the object@sc3 slot. It then creates and populates the following items of the object@sc3 slot:

• consensus - contains a list of consensus matrices. In addition to consensus matrices it also contains the Silhouette indeces of the clusters and original cell labels corresponding to the clusters.

Usage
sc3_calc_consens.SCESet(object)

## S4 method for signature 'SCESet'
sc3_calc_consens(object)

Arguments
object an object of 'SCESet' class
sc3_calc_transfs.SCESet

**Value**

an object of 'SCESet' class

---

sc3_calc_dists.SCESet *Calculate distances between the cells.*

**Description**

This function calculates distances between the cells contained in the processed_dataset item of the object@sc3 slot. It then creates and populates the following items of the object@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'
sc3_calc_dists(object)
```

**Arguments**

- object an object of 'SCESet' class

**Value**

an object of 'SCESet' class

---

sc3_calc_transfs.SCESet *Calculate transformations of the distance matrices.*

**Description**

This function calculates transformations of the distance matrices contained in the sc3_distances item of the object@sc3 slot. It then creates and populates the following items of the object@sc3 slot:

- transformations - contains a list of transformations of the distance matrices corresponding to PCA and graph Laplacian transformations.

**Usage**

```r
sc3_calc_transfs.SCESet(object)
## S4 method for signature 'SCESet'
sc3_calc_transfs(object)
```
Arguments

object  an object of 'SCESet' class

Value

an object of 'SCESet' class

---

**sc3_estimate_k.SCESet**  *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_prediction - 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.SCESet**  *Write SC3 results to Excel file*

**Description**

This function writes SC3 results contained in the object@sc3$results list 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.SCESet**

_Open SC3 results in an interactive session in a web browser_

**Description**

Runs interactive session of SC3 based on precomputed objects

**Usage**

```r
sc3_interactive. S C E S e t ( object )
```

## S4 method for signature 'SCESet'

sc3_interactive(object)

**Arguments**

object: an object of "SCESet" class

**Value**

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

---

**sc3_kmeans.SCESet**

_kmeans clustering of the transformed distance matrices._

**Description**

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

- kmeans - contains a list of kmeans clusterings.

**Usage**

```r
sc3_kmeans. S C E S e t ( object )
```

## S4 method for signature 'SCESet'

sc3_kmeans(object)

**Arguments**

object: an object of 'SCESet' class
Details

By default the nstart parameter passed to kmeans is defined in sc3_prepare.SCESet, is set to 1000 and written to kmeans_nstart item of the object@sc3 slot. If the number of cells in the dataset is more than 2000, this parameter is set to 50. A user can also manually define this parameter by changing the value of the kmeans_nstart item of the object@sc3 slot.

Value

an object of 'SCESet' class

---

sc3_plot_cell_outliers.SCESet

Plot cell outliers

Description

Outlier cells in each cluster are detected using robust distances, calculated using the minimum covariance determinant (MCD). 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 Chi-squared distribution.

Usage

sc3_plot_cell_outliers.SCESet(object, k)

## S4 method for signature 'SCESet'
sc3_plot_cell_outliers(object, k)

Arguments

object an object of 'SCESet' class
k number of clusters

---

sc3_plot_cluster_stability.SCESet

Plot stability of the clusters

Description

Stability index shows how stable each cluster is accross 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.SCESet**

*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**

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

## S4 method for signature 'SCESet'

```r
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.SCESet**

*Plot expression of DE genes of the clusters identified by SC3 as a heatmap*

**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. SC3 provides a list of all differentially expressed genes with adjusted p-values < 0.01 and plots gene expression profiles of the 50 genes with the lowest p-values. 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

`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.SCESet**

*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 cell and gene filters. 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

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

## S4 method for signature 'SCESet'

`sc3_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.SCESet

**Plot expression of marker genes of the clusters identified by SC3 as a heatmap**

**Description**
To find marker genes, for each gene a binary classifier is constructed based on the mean cluster expression values. 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. 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**
```r
sc3_plot_markers.SCESet(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.SCESet

**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**
```r
sc3_plot_silhouette.SCESet(object, k)
```

**Arguments**
- `object` an object of `SCESet` class
- `k` number of clusters
**sc3_plot_tsne.SCESet**

**Arguments**

- **object** an object of 'SCESet' class
- **k** number of clusters

**Description**

tSNE (t-Distributed Stochastic Neighbor Embedding) method is used to map high-dimensional data to a 2D space while preserving local distances between cells. tSNE has become a very popular visualisation tool. SC3 imports the Rtsne function from the Rtsne package to perform the tSNE analysis. The colors on the plot correspond to the clusters identified by SC3. One of the most sensitive parameters in tSNE analysis is the so-called perplexity. SC3 defines the default perplexity as N/5, where N is the number of cells.

**Usage**

```r
sc3_plot_tsne.SCESet(object, k,
                     perplexity = floor(ncol(get_processed_dataset(object))/5), seed = 1234567)
```

```
# S4 method for signature 'SCESet'
sc3_plot_tsne(object, k,
              perplexity = floor(ncol(get_processed_dataset(object))/5), seed = 1234567)
```

**Arguments**

- **object** an object of 'SCESet' class
- **k** number of clusters
- **perplexity** perplexity parameter used in Rtsne for tSNE transformation
- **seed** random seed used for tSNE transformation

**sc3_prepare.SCESet**

**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 object@sc3 slot:

- **processed_dataset** - contains the expression matrix to be used for SC3 clustering.
- **kmeans_iter_max** - contains a value of iter.max parameter to be used in kmeans clustering.
- **rand_seed** - contains a random seed to be used by SC3
- **kmeans_nstart** - contains a value of nstart parameter to be used in kmeans clustering.
- **n_dim** - contains values of the number of eigenvectors to be used in kmeans clustering.
- **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.
**Usage**

```r
sc3_prepare.SCESet(object, exprs_values = "exprs", gene.filter = FALSE, 
gene.filter.fraction = 0.06, gene.reads.rare = 2, gene.reads.ubiq = 0, 
log.scale = FALSE, d.region.min = 0.04, d.region.max = 0.07, 
svm.num.cells = NULL, svm.train.inds = NULL, svm.max = 5000, 
n.cores = NULL, k.means.nstart = NULL, k.means.iter.max = 1e+09, 
biology = TRUE, seed = 1)
```

```r
## S4 method for signature 'SCESet'
sc3_prepare(object, exprs_values = "exprs", 
gene.filter = FALSE, gene.filter.fraction = 0.06, gene.reads.rare = 2, 
gene.reads.ubiq = 0, log.scale = FALSE, d.region.min = 0.04, 
d.region.max = 0.07, svm.num.cells = NULL, svm.train.inds = NULL, 
svm.max = 5000, n.cores = NULL, k.means.nstart = NULL, 
k.means.iter.max = 1e+09, biology = TRUE, 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 arguments are 'tpm' (transcripts per million), 'norm_tpm' (normalised TPM values), 'fpkm' (FPKM values), 'norm_fpkm' (normalised FPKM values), 'counts' (counts for each feature), 'norm_counts', 'cpm' (counts-per-million), 'norm_cpm' (normalised counts-per-million), 'exprs' (whatever is in the 'exprs' slot of the SCESet object; default), 'norm_exprs' (normalised expression values) or 'stand_exprs' (standardised expression values) or any other named element of the assayData slot of the SCESet object that can be accessed with the get_exprs function.
- **gene.filter**: a boolean variable which defines whether to perform gene filtering before SC3 clustering. Default is TRUE. The gene filter removes genes/transcripts that are either expressed (expression value is more than gene.reads.rare) in less than X expression value is more than gene.reads.ubiq) in at least (100*X) cells (ubiquitous genes/transcripts), where X is the gene.filter.fraction*100. The motivation for the gene filter is that ubiquitous and rare genes most often are not informative for the clustering.
- **gene.filter.fraction**: fraction of cells. Default is 0.06.
- **gene.reads.rare**: expression value threshold for rare genes. Default is 2.
- **gene.reads.ubiq**: expression value threshold for ubiquitous genes. Default is 0.
- **log.scale**: a boolean variable which defines whether to perform log2 scaling before SC3 clustering. Default is TRUE.
- **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.
sc3_run_svm.SCESet

### Description

This function performs training of the SVM classifier on the training cells, which indices are contained in the `svm_train.ind` item of the object@sc3 slot. Then it predicts the labels of the remaining cells using the SVM classifier. Finally it creates and populates the following items of the object@sc3 slot:

- `svm_result` - contains labels of the cells predicted by the SVM ordered as the cells in the original dataset.

### Usage

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

### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>an object of 'SCESet' class</td>
</tr>
<tr>
<td>k</td>
<td>the number of clusters k for which the SVM should be run</td>
</tr>
</tbody>
</table>

### Value

an object of 'SCESet' class
sc3_set_ks.SCESet

Sets a range of the number of clusters k used for SC3 clustering.

Description
This function creates and populates the following items of the object@sc3 slot:

- ks - contains a range of the number of clusters k to be used by SC3

Usage
sc3_set_ks.SCESet(object, ks = NULL)

## S4 method for signature 'SCESet'
sc3_set_ks(object, ks = NULL)

Arguments

object an object of 'SCESet' class
ks a continuous range of integers - the number of clusters k used for SC3 clustering.
    Can also be a single integer.

sc3_summarise_results.SCESet

Summarise SC3 results

Description
This function summarised all SC3 results into a single list and populates it to the following item of
the object@sc3 slot:

- results - contains all SC3 results

Usage
sc3_summarise_results.SCESet(object, k)

## S4 method for signature 'SCESet'
sc3_summarise_results(object, k)

Arguments

object an object of 'SCESet' class
k        the number of clusters k for which the results should be summarised

Value
an object of 'SCESet' class
StabilityIndex

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
StabilityIndex(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{\text{sum_over_ks(\text{sum_over_clusters_N(given_cells/(given_cells + extra_cells)))}}}{\text{N(corrects for stability of each cluster)/N(corrects for the number of clusters)/length(ks)}}

Value
a numeric vector containing a stability index of each cluster

support_vector_machines

Run support vector machines (SVM) prediction

Description
Train an SVM classifier on training cells and then classify study cells using the classifier.

Usage
support_vector_machines(train, study, kern)

Arguments
- train: expression matrix with training cells
- study: expression matrix with study cells
- kern: kernel to be used with SVM

Value
classification of study cells
swatch

Plot colour swatches for a vector of colours

Description

Plot named colour swatches for a vector of colours.

Usage

swatch(x)

Arguments

x

A vector of colours, specified as: colour names (i.e. colour names returned by `colors()`); numeric indices into `palette()`, or hexadecimal strings in the form `#RRGGBB`, where RR, GG, and BB are pairs of hexadecimal digits representing red, green, and blue components, in the range 00 to FF.

Value

`NULL`. The colour swatch is plotted to the active plotting device.

See Also

`iwanthue`

tmult

Matrix left-multiplied by its transpose

Description

Given matrix A, the procedure returns A'A.

Usage

tmult(x)

Arguments

x

Numeric 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**

```r
transformation(dists, method)
```

**Arguments**

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

**Value**

transformed distance matrix

---

treutlein

**Description**

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

**Usage**

```r
treutlein
```

**Format**

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

**Source**


Columns represent cells, rows represent genes expression values. Colnames respresent 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, 25

calculate_distance, 2
consensus_matrix, 3
consmx, 3
ED1, 4
ED2, 4
estkTW, 4, 13
gene_filter, 5
gene_filter, 5
get_de_genes, 5
get_marker_cells, 6
gene_filter, 5
get_outl_cells, 6
get_processed_dataset, 7
iwanthue, 7, 24
kmeans, 10, 15, 21
kmeans, 10, 15, 21
norm_laplacian, 8
prepare_for_svm, 8
Rtsne, 19

sc3 (sc3.SCESet), 9
sc3, SCESet-method (sc3.SCESet), 9
sc3.SCESet, 9
sc3_calc_biology
  (sc3_calc_biology.SCESet), 11
sc3_calc_biology, SCESet-method
  (sc3_calc_biology.SCESet), 11
sc3_calc_biology.SCESet, 11
sc3_calc_consen
  (sc3_calc_consen.SCESet), 11
sc3_calc_consen, SCESet-method
  (sc3_calc_consen.SCESet), 11
sc3_calc_consen.SCESet, 11
sc3_calc_dists (sc3_calc_dists.SCESet), 12
sc3_calc_dists, SCESet-method
  (sc3_calc_dists.SCESet), 12
sc3_calc_dists.SCESet, 12
sc3_calc_transfs
  (sc3_calc_transfs.SCESet), 12
sc3_calc_transfs, SCESet-method
  (sc3_calc_transfs.SCESet), 12
sc3_calc_transfs.SCESet, 12
sc3_estimate_k (sc3_estimate_k.SCESet), 13
sc3_estimate_k, SCESet-method
  (sc3_estimate_k.SCESet), 13
sc3_estimate_k.SCESet, 13
sc3_export_results.xls
  (sc3_export_results.xls.SCESet), 13
sc3_export_results.xls, SCESet-method
  (sc3_export_results.xls.SCESet), 13
sc3_export_results.xls.SCESet, 13
sc3_interactive
  (sc3_interactive.SCESet), 14
sc3_interactive, SCESet-method
  (sc3_interactive.SCESet), 14
sc3_interactive.SCESet, 14
sc3_kmeans (sc3_kmeans.SCESet), 14
sc3_kmeans, SCESet-method
  (sc3_kmeans.SCESet), 14
sc3_kmeans.SCESet, 14
sc3_plot_cell_outliers
  (sc3_plot_cell_outliers.SCESet), 15
sc3_plot_cell_outliers, SCESet-method
  (sc3_plot_cell_outliers.SCESet), 15
sc3_plot_cell_outliers.SCESet, 15
sc3_plot_cluster_stability
  (sc3_plot_cluster_stability.SCESet), 15
sc3_plot_cluster_stability, SCESet-method
  (sc3_plot_cluster_stability.SCESet), 15
sc3_plot_cluster_stability.SCESet, 15
sc3_plot_consensus
  (sc3_plot_consensus.SCESet), 16
sc3_plot_consensus, SCESet-method
  (sc3_plot_consensus.SCESet), 16
sc3_plot_consensus.SCESet, 16
(sc3_plot_consensus.SCESet), 16
sc3_plot_consensus.SCESet, 16
sc3_plot_de_genes
 (sc3_plot_de_genes.SCESet), 16
sc3_plot_de_genes,SCESet-method
 (sc3_plot_de_genes.SCESet), 16
sc3_plot_de_genes.SCESet, 16
sc3_plot_expression
 (sc3_plot_expression.SCESet), 17
sc3_plot_expression,SCESet-method
 (sc3_plot_expression.SCESet), 17
sc3_plot_expression.SCESet, 17
sc3_plot_markers
 (sc3_plot_markers.SCESet), 18
sc3_plot_markers,SCESet-method
 (sc3_plot_markers.SCESet), 18
sc3_plot_markers.SCESet, 18
sc3_plot_silhouette
 (sc3_plot_silhouette.SCESet), 18
sc3_plot_silhouette,SCESet-method
 (sc3_plot_silhouette.SCESet), 18
sc3_plot_silhouette.SCESet, 18
sc3_plot_tsne (sc3_plot_tsne.SCESet), 19
sc3_plot_tsne,SCESet-method
 (sc3_plot_tsne.SCESet), 19
sc3_plot_tsne.SCESet, 19
sc3_prepare (sc3_prepare.SCESet), 19
sc3_prepare,SCESet-method
 (sc3_prepare.SCESet), 19
sc3_prepare.SCESet, 19
sc3_run_svm (sc3_run_svm.SCESet), 21
sc3_run_svm,SCESet-method
 (sc3_run_svm.SCESet), 21
sc3_run_svm.SCESet, 21
sc3_set_ks (sc3_set_ks.SCESet), 22
sc3_set_ks,SCESet-method
 (sc3_set_ks.SCESet), 22
sc3_set_ks.SCESet, 22
sc3_summarise_results
 (sc3_summarise_results.SCESet), 22
sc3_summarise_results,SCESet-method
 (sc3_summarise_results.SCESet), 22
sc3_summarise_results.SCESet, 22
StabilityIndex, 23
support_vector_machines, 23
swatch, 7, 8, 24
tmult, 24
transformation, 25
treutlein, 25