Package 'scmap'

May 8, 2024

Type Package

Title A tool for unsupervised projection of single cell RNA-seq data

Version 1.26.0

Author Vladimir Kiselev

Maintainer Vladimir Kiselev <vladimir.yu.kiselev@gmail.com>

Description Single-cell RNA-seq (scRNA-seq) is widely used to investigate the composition of complex tissues since the technology allows researchers to define cell-types using unsupervised clustering of the transcriptome. However, due to differences in experimental methods and computational analyses, it is often challenging to directly compare the cells identified in two different experiments. scmap is a method for projecting cells from a scRNA-seq experiment on to the cell-types or individual cells identified in a different experiment.

License GPL-3

Imports Biobase, SingleCellExperiment, SummarizedExperiment, BiocGenerics, S4Vectors, dplyr, reshape2, matrixStats, proxy, utils, googleVis, ggplot2, methods, stats, e1071, randomForest, Rcpp (>= 0.12.12)

Depends R(>= 3.4)

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, BiocStyle

VignetteBuilder knitr

biocViews ImmunoOncology, SingleCell, Software, Classification, SupportVectorMachine, RNASeq, Visualization, Transcriptomics, DataRepresentation, Transcription, Sequencing, Preprocessing, GeneExpression, DataImport

NeedsCompilation no

URL https://github.com/hemberg-lab/scmap

BugReports https://support.bioconductor.org/t/scmap/
git_url https://git.bioconductor.org/packages/scmap
git_branch RELEASE_3_19
git_last_commit 5629783
git_last_commit_date 2024-04-30
Repository Bioconductor 3.19
Date/Publication 2024-05-07

Contents

---

ann *Cell type annotations for data extracted from a publication by Yan et al.*

Description

Cell type annotations for data extracted from a publication by Yan et al.

Usage

```
ann
```

Format

An object of class `data.frame` with 90 rows and 1 columns.

Source

http://dx.doi.org/10.1038/nsmb.2660

Each row corresponds to a single cell from ‘yan‘ dataset

`EuclSqNorm` *The Euclidean Squared Norm of each column of a matrix is computed and the whole result is returned as a vector. Used as part of the approx. calculations of the cosine similarity between the query and the reference.*

Description

The Euclidean Squared Norm of each column of a matrix is computed and the whole result is returned as a vector. Used as part of the approx. calculations of the cosine similarity between the query and the reference.

Usage

```
EuclSqNorm(dat)
```

Arguments

`dat`	A numerical matrix

`getSankey` *Plot Sankey diagram comparing two clusterings*

Description

Sometimes it is useful to see how the clusters in two different clustering solutions correspond to each other. Sankey diagram is a good way to visualize them. This function takes as input two clustering solutions and visualizes them using a Sankey diagram. The order of the reference clusters is defined by their labels in increasing order.

Usage

```
getSankey(reference, clusters, plot_width = 400, plot_height = 600,
  colors = NULL)
```

Arguments

`reference`	reference clustering labels
`clusters`	clustering labels under investigations
`plot_width`	width of the output plot in pixels
`plot_height`	height of the output plot in pixels
`colors`	colors of the links between two clusterings. If defined please note that each cluster in the reference clustering has to have its own color. This should be a normal text vector, e.g. c('#FF0000', '#FFA500', '#008000')

Value

an object returned by ‘gvisSankey‘

Examples

```
plot(getSankey(ann[ , 1], ann[ , 1]))
```

`indexCell` *Create an index for a dataset to enable fast approximate nearest neighbour search*

Description

The method is based on product quantization for the cosine distance. Split the training data into M identically sized chunks by genes. Use k-means to find k subcentroids for each group. Assign cluster numbers to each member of the dataset.

Usage

```
indexCell(object = NULL, M = NULL, k = NULL)

indexCell.SingleCellExperiment(object, M, k)

## S4 method for signature 'SingleCellExperiment'
indexCell(object = NULL, M = NULL,
  k = NULL)
```

Arguments

`object`	an object of `SingleCellExperiment` class
`M`	number of chunks into which the expr matrix is split
`k`	number of clusters per group for k-means clustering

Value

a list of four objects: 1) a list of matrices containing the subcentroids of each group 2) a matrix containing the subclusters for each cell for each group 3) the value of M 4) the value of k

Examples

```
library(SingleCellExperiment)
sce <- SingleCellExperiment(assays = list(normcounts = as.matrix(yan)), colData = ann)
# this is needed to calculate dropout rate for feature selection
# important: normcounts have the same zeros as raw counts (fpkm)
counts(sce) <- normcounts(sce)
logcounts(sce) <- log2(normcounts(sce) + 1)
# use gene names as feature symbols
```

```
rowData(sce)$feature_symbol <- rownames(sce)
# remove features with duplicated names
sce <- sce[!duplicated(rownames(sce)), ]
sce <- selectFeatures(sce)
sce <- indexCell(sce)
```

`indexCluster` *Create a precomputed Reference*

Description

Calculates centroids of each cell type and merge them into a single table.

Usage

```
indexCluster(object = NULL, cluster_col = "cell_type1")

indexCluster.SingleCellExperiment(object, cluster_col)

## S4 method for signature 'SingleCellExperiment'
indexCluster(object = NULL,
  cluster_col = "cell_type1")
```

Arguments

`object`	SingleCellExperiment object
`cluster_col`	column name in the ‘colData‘ slot of the SingleCellExperiment object containing the cell classification information

Value

a ‘data.frame‘ containing calculated centroids of the cell types of the Reference dataset

Examples

```
library(SingleCellExperiment)
sce <- SingleCellExperiment(assays = list(normcounts = as.matrix(yan)), colData = ann)
# this is needed to calculate dropout rate for feature selection
# important: normcounts have the same zeros as raw counts (fpkm)
counts(sce) <- normcounts(sce)
logcounts(sce) <- log2(normcounts(sce) + 1)
# use gene names as feature symbols
rowData(sce)$feature_symbol <- rownames(sce)
# remove features with duplicated names
sce <- sce[!duplicated(rownames(sce)), ]
sce <- selectFeatures(sce)
sce <- indexCluster(sce[rowData(sce)$scmap_features, ])
```

NN *Main nearest neighbour calculation function. Used on the first reference dataset. Returns a list of three objects: 1) the cell indices of the w nearest neighbours 2) the corresponding approx. cosine similarities*

Description

Main nearest neighbour calculation function. Used on the first reference dataset. Returns a list of three objects: 1) the cell indices of the w nearest neighbours 2) the corresponding approx. cosine similarities

Usage

```
NN(w, k, subcentroids, subclusters, query_chunks, M, SqNorm)
```

Arguments

w	An integer specifying the number of nearest neighbours
k	An integer specifying the number of subcentroids for each product quantization chunk
subcentroids	A list of matrices containing the subcentroids of each chunk.
subclusters	A matrix containing the subcentroid assignments of each reference cell. See scf_index.
query_chunks	A list of matrices containing the chunks of the query dataset after it has been split according to the product quantization method
M	An integer specifying the number of chunks
SqNorm	A numerical vector containing the Euclidean Squared Norm of each query cell.

normalise *Normalises each column of a matrix*

Description

Normalises each column of a matrix

Usage

```
normalise(dat)
```

Arguments

dat	A numerical matrix

scmapCell *For each cell in a query dataset, we search for the nearest neighbours by cosine distance within a collection of reference datasets.*

Description

For each cell in a query dataset, we search for the nearest neighbours by cosine distance within a collection of reference datasets.

Usage

```
scmapCell(projection = NULL, index_list = NULL, w = 10)

scmapCell.SingleCellExperiment(projection, index_list, w)

## S4 method for signature 'SingleCellExperiment'
scmapCell(projection = NULL,
  index_list = NULL, w = 10)
```

Arguments

projection	an object of SingleCellExperiment class
index_list	list of index objects each coming from the output of 'indexCell'
w	a positive integer specifying the number of nearest neighbours to find

Value

a list of 3 objects: 1) a matrix with the closest w neighbours by cell number of each query cell stored by column 2) a matrix of integers giving the reference datasets from which the above cells came from 3) a matrix with the cosine similarities corresponding to each of the nearest neighbours

Examples

```
library(SingleCellExperiment)
sce <- SingleCellExperiment(assays = list(normcounts = as.matrix(yan)), colData = ann)
# this is needed to calculate dropout rate for feature selection
# important: normcounts have the same zeros as raw counts (fpkm)
counts(sce) <- normcounts(sce)
logcounts(sce) <- log2(normcounts(sce) + 1)
# use gene names as feature symbols
rowData(sce)$feature_symbol <- rownames(sce)
# remove features with duplicated names
sce <- sce[!duplicated(rownames(sce)), ]
sce <- selectFeatures(sce)
sce <- indexCell(sce)
scmapCell_results <- scmapCell(sce, list(metadata(sce)$scmap_cell_index))
```

scmapCell2Cluster *Approximate k-NN cell-type classification using scfinemap*

Description

Each cell in the query dataset is assigned a cell-type if the similarity between its nearest neighbour exceeds a threshold AND its w nearest neighbours have the same cell-type.

Usage

```
scmapCell2Cluster(scmapCell_results = NULL, cluster_list = NULL, w = 3,
  threshold = 0.5)

scmapCell2Cluster.SingleCellExperiment(scmapCell_results, cluster_list, w,
  threshold)

## S4 method for signature 'list'
scmapCell2Cluster(scmapCell_results = NULL,
  cluster_list = NULL, w = 3, threshold = 0.5)
```

Arguments

scmapCell_results	the output of ‘scmapCell()‘ with ‘projection‘ as its input.
cluster_list	list of cell cluster labels correspondint to each index against which the ‘projection‘ has been projected
w	an integer specifying the number of nearest neighbours to find
threshold	the threshold which the maximum similarity between the query and a reference cell must exceed for the cell-type to be assigned

Value

The query dataset with the predicted labels attached to colData(query_dat)$cell_type1

Examples

```
library(SingleCellExperiment)
sce <- SingleCellExperiment(assays = list(normcounts = as.matrix(yan)), colData = ann)
# this is needed to calculate dropout rate for feature selection
# important: normcounts have the same zeros as raw counts (fpkm)
counts(sce) <- normcounts(sce)
logcounts(sce) <- log2(normcounts(sce) + 1)
# use gene names as feature symbols
rowData(sce)$feature_symbol <- rownames(sce)
# remove features with duplicated names
sce <- sce[!duplicated(rownames(sce)), ]
sce <- selectFeatures(sce)
```

```
sce <- indexCell(sce)
scmapCell_results <- scmapCell(sce, list(metadata(sce)$scmap_cell_index))
sce <- scmapCell2Cluster(scmapCell_results, cluster_list = list(colData(sce)$cell_type1))
```

---

scmapCluster *scmap main function*

Description

Projection of one dataset to another

Usage

```
scmapCluster(projection = NULL, index_list = NULL, threshold = 0.7)

scmapCluster.SingleCellExperiment(projection, index_list, threshold)

## S4 method for signature 'SingleCellExperiment'
scmapCluster(projection = NULL,
  index_list = NULL, threshold = 0.7)
```

Arguments

projection	'SingleCellExperiment' object to project
index_list	list of index objects each coming from the output of 'indexCluster'
threshold	threshold on similarity (or probability for SVM and RF)

Value

The projection object of SingleCellExperiment class with labels calculated by 'scmap' and stored in the scmap_labels column of the rowData(object) slot.

Examples

```
library(SingleCellExperiment)
sce <- SingleCellExperiment(assays = list(normcounts = as.matrix(yan)), colData = ann)
# this is needed to calculate dropout rate for feature selection
# important: normcounts have the same zeros as raw counts (fpkm)
counts(sce) <- normcounts(sce)
logcounts(sce) <- log2(normcounts(sce) + 1)
# use gene names as feature symbols
rowData(sce)$feature_symbol <- rownames(sce)
# remove features with duplicated names
sce <- sce[!duplicated(rownames(sce)), ]
sce <- selectFeatures(sce)
sce <- indexCluster(sce)
sce <- scmapCluster(sce, list(metadata(sce)$scmap_cluster_index))
```

`selectFeatures` *Find the most informative features (genes/transcripts) for projection*

Description

This is a modification of the M3Drop method. Instead of fitting a Michaelis-Menten model to the log expression-dropout relation, we fit a linear model. Namely, the linear model is build on the log(expression) versus log(dropout) distribution. After fitting a linear model important features are selected as the top N residuals of the linear model.

Usage

```
selectFeatures(object, n_features = 500, suppress_plot = TRUE)

selectFeatures.SingleCellExperiment(object, n_features, suppress_plot)

## S4 method for signature 'SingleCellExperiment'
selectFeatures(object, n_features = 500,
  suppress_plot = TRUE)
```

Arguments

`object`	an object of `SingleCellExperiment` class
`n_features`	number of the features to be selected
`suppress_plot`	boolean parameter, which defines whether to plot log(expression) versus log(dropout) distribution for all genes. Selected features are highlighted with the red colour.

Details

Please note that `feature_symbol` column of `rowData(object)` must be present in the input object and should not contain any duplicated feature names. This column defines feature names used during projection. Feature symbols in the reference dataset must correpond to the feature symbols in the projection dataset, otherwise the mapping will not work!

Value

an object of `SingleCellExperiment` class with a new column in `rowData(object)` slot which is called `scmap_features`. It can be accessed by using `as.data.frame(rowData(object))$scmap_features`.

Examples

```
library(SingleCellExperiment)
sce <- SingleCellExperiment(assays = list(normcounts = as.matrix(yan)), colData = ann)
# this is needed to calculate dropout rate for feature selection
# important: normcounts have the same zeros as raw counts (fpkm)
counts(sce) <- normcounts(sce)
logcounts(sce) <- log2(normcounts(sce) + 1)
```

```
# use gene names as feature symbols
rowData(sce)$feature_symbol <- rownames(sce)
# remove features with duplicated names
sce <- sce[!duplicated(rownames(sce)), ]
sce <- selectFeatures(sce)
```

`setFeatures` *Set the most important features (genes/transcripts) for projection*

Description

This method manually sets the features to be used for projection.

Usage

```
setFeatures(object, features = NULL)

setFeatures.SingleCellExperiment(object, features)

## S4 method for signature 'SingleCellExperiment'
setFeatures(object, features = NULL)
```

Arguments

`object`	an object of `SingleCellExperiment` class
`features`	a character vector of feature names

Details

Please note that `feature_symbol` column of `rowData(object)` must be present in the input object and should not contain any duplicated feature names. This column defines feature names used during projection. Feature symbols in the reference dataset must correpond to the feature symbols in the projection dataset, otherwise the mapping will not work!

Value

an object of `SingleCellExperiment` class with a new column in `rowData(object)` slot which is called `scmap_features`. It can be accessed by using `as.data.frame(rowData(object))$scmap_features`.

Examples

```
library(SingleCellExperiment)
sce <- SingleCellExperiment(assays = list(normcounts = as.matrix(yan)), colData = ann)
# this is needed to calculate dropout rate for feature selection
# important: normcounts have the same zeros as raw counts (fpkm)
counts(sce) <- normcounts(sce)
logcounts(sce) <- log2(normcounts(sce) + 1)
```

```
# use gene names as feature symbols
rowData(sce)$feature_symbol <- rownames(sce)
# remove features with duplicated names
sce <- sce[!duplicated(rownames(sce)), ]
sce <- setFeatures(sce, c('MMP2', 'ZHX3'))
```

`subdistsmult` *Computes the dot product between the subcentroids from the indexed reference and the subvectors of an element of the query dataset. Returns an M by k matrix. Used as an intermediate step (in NNfirst and NNmult) for calculating an approximation of the cosine similarity between the query and the reference.*

Description

Computes the dot product between the subcentroids from the indexed reference and the subvectors of an element of the query dataset. Returns an M by k matrix. Used as an intermediate step (in NNfirst and NNmult) for calculating an approximation of the cosine similarity between the query and the reference.

Usage

```
subdistsmult(subcentroids, query_chunks, M, k, cellnum)
```

Arguments

`subcentroids`	A list of matrices containing the subcentroids of each chunk.
`query_chunks`	A list of matrices containing the chunks of the query dataset after it has been split according to the product quantization method
`M`	An integer specifying the number of chunks
`k`	An integer specifying the number of subcentroids per chunk
`cellnum`	An integer specifying the column of the query dataset we wish to consider

---

`yan` *Single cell RNA-Seq data extracted from a publication by Yan et al.*

Description

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

Usage

```
yan
```

Format

An object of class `data.frame` with 20214 rows and 90 columns.

Source

http://dx.doi.org/10.1038/nsmb.2660

Columns represent cells, rows represent genes expression values.

Index