Package 'screamr'

December 26, 2019

Type Package

Title An R package for scRNA-Seq data analysis

Version 0.1

Date 2019-12-26

Author Guilherme Sena

Maintainer Guilherme Sena <desenabr@usc.edu>

Description An R package for scRNA-Seq data analysis

License GNU Public License v3.0

Imports Rcpp (>= 1.0.3),
RMTstat,
mclust

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.0.2

R topics documented:

`screamr-package` *An R package for scRNA-Seq data analysis*

Description

An R package for scRNA-Seq data analysis

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.
~~ An overview of how to use the package, including the most important ~~ ~~ functions ~~

Author(s)

Guilherme Sena

Maintainer: Guilherme Sena <desenabr@usc.edu>

References

~~ Literature or other references for background information ~~

See Also

~~ Optional links to other man pages, e.g. ~~ ~~ `<pkg>` ~~

Examples

```
~~ simple examples of the most important functions ~~
```

ClusterCells *Clusters the single cells using gaussian mixture model*

Description

Runs Mclust the way it is supposed to be used in the scRNA-Seq context

Usage

```
ClusterCells(m.svd, N = 30, verbose = T)
```

Arguments

m.svd	The reduced dimension dataset, rows as cells columns as PCs
N	(temporary) the maximum number of clusters to try

Value

an mclust object with the probabilistic clustering results

An mclust object with clustering results: Classification, probabilities for each cell and mean/variance/probability parameters for each cluster.

Examples

```
m.raw <- matrix(rpois(1000, lambda = 10), ncol = 20)
m <- LogNormalize(m)
m.scale <- GeneScale(m)
m.svd <- ReduceDimenson(m.scale)
clusters <- ClusterCells(m.svd)
```

ClusterSeparation *Calcualtes a measure of separability*

Description

Function to calculate cluster separation, ie, the ratio between the variance within clusters and the variance between clusters. A smaller value means the cluster is more separated. Note that separation > 1 means that the variance within clusters is higher than the variance between, and this is an indication that the dataset was overclustered.

Usage

```
ClusterSeparation(m.svd, clusters)
```

Arguments

m.svd	the reduced dimension form the ReduceDimension function
clusters	the mclust object with probabilistic cluster assignments

Value

A value between 0 (clusters fully separated) and infinity

Examples

```
m.raw <- matrix(rpois(1000, lambda = 10), ncol = 20)
m <- LogNormalize(m)
m.scale <- GeneScale(m)
m.svd <- ReduceDimenson(m.scale)
clusters <- ClusterCells(m.svd)
ClusterSeparation(m.svd,clusters)
```

ClusterWithFixedK *Clusters the single cells using gaussian mixture model for a fixed value of k*

Description

Clusters the single cells using gaussian mixture model for a fixed value of k

Usage

```
ClusterWithFixedK(m.svd, k, init = NULL, verbose = T)
```

Arguments

m.svd	The reduced dimension dataset, rows as cells columns as PCs
k	The number of clusters to run EM on
init	the Mclust::hc initialization function

Value

an mclust object with the probabilistic clustering results

An mclust object with clustering results: Classification, probabilities for each cell and mean/variance/probability parameters for each cluster.

Examples

```
m.raw <- matrix(rpois(1000, lambda = 10), ncol = 20)
m <- LogNormalize(m)
m.scale <- GeneScale(m)
m.svd <- ReduceDimenson(m.scale)
clusters <- ClusterWithFixedK(m.svd, k = 3)
```

`ColSums` *Sum columns of a sparse matrix*

Description

Sum columns of a sparse matrix

Usage

```
ColSums(m)
```

`ColumnScale` *Converts count data into RPX*

Description

This function attempts to size factor normalize the raw counts. If no genes are expressed across all cells, size factors are approximated by library size.

Usage

```
ColumnScale(m.raw, size.factors = T, norm.factor = NULL)
```

Arguments

`m.raw`	the raw data matrix
`size.factors`	optional, if you know that size factors cannot be estimated, size factor detection can be skipped

Value

The log-normalized matrix

Examples

```
m.raw <- matrix(rpois(1000, lambda = 10), ncol = 20)
m <- LogNormalize(m)
```

`DiffExpr` *Student's t-test to find differentially expressed genes on each cluster*

Description

This function uses the scaled values calculated by GeneScale to run student's t-test and find genes that mark certain clusters

Usage

```
DiffExpr(m.scale, clusters, cl1, cl2)
```

Arguments

`m.scale`	the N x M scaled normalized count matrix given by GeneScale
`clusters`	the clustering result from ClusterCells with k clusters
`which.cluster`	the cluster to be tested

Value

An N x (2k - 2) table, showing the log fold change of each gene in the tested cluster vs all other cluster, with respective p-values

Examples

```
m.raw <- matrix(rpois(1000, lambda = 10), ncol = 20)
m <- LogNormalize(m)
m.scale <- GeneScale(m)
m.svd <- ReduceDimenson(m.scale)
clusters <- ClusterCells(m.svd)
# Finds DE genes for cluster 1 vs 2
tbl <- DiffExpr(m.scale, clusters$classification, 1,2)
```

`DiffExprAll` *Student's t-test to find differentially expressed genes on each cluster*

Description

This function uses the scaled values calculated by GeneScale to run student's t-test and find genes that mark certain clusters

Usage

```
DiffExprAll(m.scale, clusters, which.cluster)
```

Arguments

m.scale	the N x M scaled normalized count matrix given by GeneScale
clusters	the clustering result from ClusterCells with k clusters
which.cluster	the cluster to be tested

Value

An N x (2k - 2) table, showing the log fold change of each gene in the tested cluster vs all other cluster, with respective p-values

Examples

```
m.raw <- matrix(rpois(1000, lambda = 10), ncol = 20)
m <- LogNormalize(m)
m.scale <- GeneScale(m)
m.svd <- ReduceDimenson(m.scale)
clusters <- ClusterCells(m.svd)
# Finds DE genes for cluster 1
tbl <- DiffExprAll(m.scale, clusters$classification, 1)
```

---

estimateSizeFactorsForMatrix *reimplementing size factor estimation to avoid DESeq dependency*

Description

reimplementing size factor estimation to avoid DESeq dependency

Usage

```
estimateSizeFactorsForMatrix(counts, locfunc = stats::median)
```

GeneScale *Gene-wise scaling of log-normalized data*

Description

Subtracts the row mean and divides by the row standard deviation.

Usage

```
GeneScale(m)
```

Arguments

m	the normalized matrix (eg: from LogNormalize)

Value

The scaled matrix, where each row has mean = 0 and sd = 1

Examples

```
m.raw <- matrix(rpois(1000, lambda = 10), ncol = 20)
m <- LogNormalize(m)
m.scale <- GeneScale(m)
```

GetMarkers *Finds marker genes from differential expression results*

Description

Given a test from DiffExpr, returns the genes which are significantly larger in a cluster vs all other clusters, these are considered to be marker genes.

Usage

```
GetMarkers(diffexpr, sig.level = 0.05)
```

Arguments

`diffexpr`	the result from the DiffExpr function
`sig.level`	the maximum p-value to be considered significantly DE. Genes where p < sig.level in every cluster will be returned.

Value

A list of genes given by the m.scale row names

Examples

```
m.raw <- matrix(rpois(1000, lambda = 10), ncol = 20)
m <- LogNormalize(m)
m.scale <- GeneScale(m)
m.svd <- ReduceDimenson(m.scale)
clusters <- ClusterCells(m.svd)
# Finds DE genes for cluster 1
tbl <- DiffExpr(m.scale, clusters$classification, 1)
# Finds the markers of cluster 1 with p < 0.05 on all tests
markers <- GetMarkers(tbl, sig.level = 0.05)
```

`GroupByFeature` *Averages the reduced dimension dataset by a given feature*

Description

Averages the reduced dimension dataset by a given feature

Usage

```
GroupByFeature(m, group, func = sum)
```

Arguments

`m`	the N x M count matrix
`group`	the vector of length M with k factors

Value

the N x k matrix with cells grouped by the group feature (eg: sum or mean)

`GuessPhenotype` *Given a set of markers, guesses the phenotype based on smallest p value of a set of hypergeometric tests from a phenotype list*

Description

Given a set of markers, guesses the phenotype based on smallest p value of a set of hypergeometric tests from a phenotype list

Usage

```
GuessPhenotype(markers, all.genes, pheno.list, verbose = T)
```

`HyperTest` *Hypergeometric test if marker genes significantly resemble a given phenotype*

Description

Outputs the probability that a set of K marker genes have k genes in common with a set of n phenotype genes in a universe of N total genes

Usage

```
HyperTest(all.genes, pheno.genes, marker.genes)
```

Arguments

`all.genes`	a set of N genes representing the ones tested for DE
`pheno.genes`	a set of n genes, contained in all genes, that represent some phenotype
`marker.genes`	a set of K genes, contained in all.genes

Value

A list of genes given by the m.scale row names

Examples

```
m.raw <- matrix(rpois(1000, lambda = 10), ncol = 20)
m <- LogNormalize(m)
m.scale <- GeneScale(m)
m.svd <- ReduceDimenson(m.scale)
clusters <- ClusterCells(m.svd)
# Finds DE genes for cluster 1
tbl <- DiffExpr(m.scale, clusters$classification, 1)
# Finds the markers of cluster 1 with p < 0.05 on all tests
markers <- GetMarkers(tbl, sig.level = 0.05)
```

`InsertIntoDatabase` *Function to insert into the database*

Description

This function takes as an input an mtx file, analyzes it and inserts into the database with all centroids concatenated and

Usage

```
InsertIntoDatabase(
  IN.MTX.RAW.FILE,
  IN.GENES.TSV.FILE,
  DATABASE.PATH,
  srp = NULL,
  srr = NULL,
  sample.name = NULL
)
```

Arguments

`IN.MTX.RAW.FILE`	the raw count data from an mtx file
`IN.GENES.TSV.FILE`	the gene names of each row in the mtx file
`DATABASE.PATH`	the directory path to which the analysis should be saved in
`srp`	the Sequencing Read Project (SRP) for the dataset
`srr`	the Sequencing Read Run (SRR) for the dataset
`sample.name`	An optional name for the dataset

`MakePathwayMatrix` *Makes an expression matrix of pathways*

Description

Makes an expression matrix of pathways

Usage

```
MakePathwayMatrix(count.mtx, pathway.list, verbose = F)
```

Arguments

`count.mtx`	the n x m normalized matrix (eg from GeneScale)
`pathway.list`	the list file with k pathways as names and gene lists as
`verbose`	print more run info values, eg: from ReadPathwaysIntoList function

Value

a k x m matrix with pathway scores for each cell

`MtxFixGenes` *Converts mtx file into a different set of gene symbols Genes that do not originally exist in the matrix are treated as zeros across all cells*

Description

Converts mtx file into a different set of gene symbols Genes that do not originally exist in the matrix are treated as zeros across all cells

Usage

```
MtxFixGenes(mtx, new.genes)
```

`PlotGene` *Plots a dataset gene in reduced dimension*

Description

Plots a gene in the reduced dimension database res = any 2D dimension reduction (svd, tsne, umap) samples = the discrete values to color points by

Usage

```
PlotGene(
  res,
  m.scale,
  gene = NULL,
  symbols = NULL,
  file = NULL,
  xlab = NULL,
  ylab = NULL,
  colors = c("lightgray", "blue"),
  width = 16,
  height = 9
)
```

Arguments

res	the N x 2 reduced dimension, where N is the number of cells
m.scale	the scaled matrix from GeneScale, the values of which will be used to color the points by gene expression
gene	the gene to plot
file	optional, a path to save the plot as a pdf file, if provided

Examples

```
library(umap)
um <- umap(m.svd)
SCREAM.PlotGene(um$layout, m.scale, "Cd24")
```

PlotReduction *Plots the dataset in reduced dimension*

Description

Plots the reduced dimension database res = any 2D dimension reduction (svd, tsne, umap) samples = the discrete values to color points by

Usage

```
PlotReduction(
  res,
  samples = NULL,
  file = NULL,
  symbols = NULL,
  points = T,
  centroids = T,
  width = 16,
  height = 9,
  main = "",
  edges = NULL,
  edge.colors = NULL,
  plot.colors = SCREAM.COLORS
)
```

Arguments

res	the N x 2 reduced dimension, where N is the number of cells
samples	optional, the samples to color by
file	optional, a file path to save the plot as a pdf file, if provided
points	if false, points will become the first letter of the sample name. Used when there are too many samples to color
main	optional, the title of the plot
edges	optional, if provided, connects the sample centroids by edges, often used when representing transitions between clusters within the plot
plot.colors	optional, the color palette to use to color points.

Examples

```
library(umap)
um <- umap(m.svd)
PlotReduction(um$layout, samples = cluster$classification)
```

RcppEigen-Functions

Set of functions in example RcppEigen package

Description

These four functions are created when `RcppEigen.package.skeleton()` is invoked to create a skeleton packages.

Usage

```
rcppeigen_hello_world()
rcppeigen_outerproduct(x)
rcppeigen_innerproduct(x)
rcppeigen_bothproducts(x)
```

Arguments

x	a numeric vector

Details

These are example functions which should be largely self-explanatory. Their main benefit is to demonstrate how to write a function using the Eigen C++ classes, and to have to such a function accessible from R.

Value

`rcppeigen_hello_world()` does not return a value, but displays a message to the console.

`rcppeigen_outerproduct()` returns a numeric matrix computed as the outer (vector) product of `x`.

`rcppeigen_innerproduct()` returns a double computer as the inner (vector) product of `x`.

`rcppeigen_bothproducts()` returns a list with both the outer and inner products.

Author(s)

Dirk Eddelbuettel

References

See the documentation for Eigen, and RcppEigen, for more details.

Examples

```
x <- sqrt(1:4)
rcppeigen_innerproduct(x)
rcppeigen_outerproduct(x)
```

Read10X *Reads 10X output matrix*

Description

Reads 10X output matrix

Usage

```
Read10X(dir)
```

ReadCellMarkersIntoList *Reads file downloaed from http://bio-bigdata.hrbmu.edu.cn/CellMarker into a list of phenotypes, whose names are phenotypes and values are vectors marker genes*

Description

Reads file downloaed from http://bio-bigdata.hrbmu.edu.cn/CellMarker into a list of phenotypes, whose names are phenotypes and values are vectors marker genes

Usage

```
ReadCellMarkersIntoList(markers.file)
```

Arguments

the	filename of a CellMarker table, eg /path/to/db/mm10/metadata/markers.tsv

Value

a list with phenotypes as names and genes as values

`readMM` *Reads mtx path as dgCMatrix*

Description

Reads mtx path as dgCMatrix

Usage

```
readMM(mtx.file)
```

`ReadPanglaoIntoList` *Reads file downloaded from https://panglaodb.se/markers.html into a list of phenotypes, whose names are phenotypes and values are vectors marker genes*

Description

Reads file downloaded from https://panglaodb.se/markers.html into a list of phenotypes, whose names are phenotypes and values are vectors marker genes

Usage

```
ReadPanglaoIntoList(markers.file)
```

Arguments

`the`	filename of a Panglao table, eg /path/to/db/mm10/metadata/panglao.tsv

Value

a list with phenotypes as names and genes as values

`ReduceDimension` *Reduces the matrix to SVD space*

Description

Performs svd and keeps only the dimensions whose eigenvector is larger than the theoretical maximum of the same matrix with standard normal values

Usage

```
ReduceDimension(m.scale)
```

Arguments

`m.scale`	the scaled matrix (eg: from GeneScale)

Value

The matrix product V * Sigma in the Singular Value Decomposition of m.scale

Examples

```
m.raw <- matrix(rpois(1000, lambda = 10), ncol = 20)
m <- LogNormalize(m)
m.scale <- GeneScale(m)
m.svd <- ReduceDimenson(m.scale)
```

ReduceDimensionTruncated

Reduces the matrix to SVD space to a fixed number of dimensions

---

Description

Performs svd and keeps only the dimensions whose eigenvector is larger than the theoretical maximum of the same matrix with standard normal values. Unlike ReduceDimension, this function is a faster version when a 'guess' on number of eigenvalues is given. If the number of significant eigenvalues is smaller than the guess, then this function should be used with significant performance and no error, otherwise a warning is printed to increase the guess size.

Usage

```
ReduceDimensionTruncated(m.scale, nv = 100)
```

Arguments

m.scale	the scaled matrix (eg: from GeneScale)

Value

The matrix product V * Sigma in the Singular Value Decomposition of m.scale

Examples

```
m.raw <- matrix(rpois(1000, lambda = 10), ncol = 20)
m <- LogNormalize(m)
m.scale <- GeneScale(m)
m.svd <- ReduceDimensonTruncated(m.scale, N = 100)
```

RowSums

Sum rows of a sparse matrix

Description

Sum rows of a sparse matrix

Usage

```
RowSums(m)
```

SCREAM.CoNormalize *Database conormalization*

Description

This function takes as an input the centroids from the database and the metadata relative to each centroid and co-normalizes them based on phenotype.

Usage

```
SCREAM.CoNormalize(db, metadata)
```

Arguments

db	the database mtx, with columns as database centroids and rows as genes.
metadata	the metadata read from the SCREAM database, with columns as SRP, SRR, phenotype and additional observations

Value

The qsmooth co-normalized matrix

WithinBetweenRatio *Ratio between distance of matching clusters and distance between clusters within the same dataset*

Description

Ratio between distance of matching clusters and distance between clusters within the same dataset

Usage

```
WithinBetweenRatio(dist.matrix, pheno.dataset.tbl)
```

Arguments

dist.matrix	the pairwise distance between clusters
pheno.dataset.tbl	a table with two columns: phenotype and dataset of origin

Value

the k x d averages of cells from each group

Index