Package 'DUBStepR'

October 5, 2021

Type Package

Title Correlation-Based Feature Selection for Single-Cell RNA Sequencing Data

Version 1.2.0

Maintainer Bobby Ranjan <ranjan_bobby@gis.a-star.edu.sg>

Description Determining the optimal set of feature genes to characterise cell types in single-cell RNA sequencing data using stepwise regression on gene-gene correlations. <doi:10.1101/2020.10.07.330563>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports Matrix, matrixcalc, RANN, qlcMatrix, parallel, stats, Seurat, methods, graphics

RoxygenNote 7.1.1

Depends R (>= 3.5.0)

Suggests rmarkdown, knitr, hdf5r, dplyr

VignetteBuilder knitr

NeedsCompilation no

Author Bobby Ranjan [aut, cre],
Kunal Mishra [ctb],
Shyam Lab [cph],
Bryan Hanson [ctb],
Damian Coventry [ctb],
Brandon Crosby [ctb],
Gregoire Thomas [ctb]

Repository CRAN

Date/Publication 2021-10-05 11:00:02 UTC

R topics documented:

`DUBStepR` *DUBStepR - Obtain a list of feature genes to characterise cell types*

Description

DUBStepR - Obtain a list of feature genes to characterise cell types

Usage

```
DUBStepR(
  input.data,
  min.cells = 0.05 * ncol(input.data),
  species = "human",
  optimise.features = TRUE,
  k = 10,
  num.pcs = 20,
  error = 0
)
```

Arguments

`input.data` input gene expression matrix (genes x cells)

`min.cells` minimum number of cells to filter genes out

`species` species to use for gene filtering: "human" (default), "mouse" and "rat"

`optimise.features` Determine optimal feature set using density index

`k` number of nearest neighbours. Default is 10

`num.pcs` number of principal components to represent sc data. Default is 20

`error` Acceptable error margin for kNN computation. Default is 0, but is set to 1 for large datasets

Value

Returns optimal feature set

Author(s)

ranjanb

Examples

```
res <- DUBStepR(input.data = pbmc_norm_small_data)
```

---

findElbow *Find the Elbow in a Curve*

Description

This utility function finds the elbow in a curve which is concave relative to a line drawn between the first and last points. The elbow is defined as the point with the greatest orthogonal distance from that line.

Usage

```
findElbow(y, ylab = "y values", plot = FALSE, returnIndex = TRUE)
```

Arguments

`y`	Numeric vector of y values for the curve.
`ylab`	Y-axis label.
`plot`	Logical. Should a plot be made?
`returnIndex`	Logical. Should the return value be the index of the elbow point?

Value

If `returnIndex = TRUE`, the index of the elbow point. If `returnIndex = FALSE`, a data frame containing an index values (x), the y values passed to the function, and the the orthogonal distances of the y values from the line connecting the first and last points. `which.max(data_frame_name$dist)` will give the index of the elbow point.

Warning

This function makes some simple checks that the data is concave as defined above. Even so, it may give answers in some cases that are not valid. Please check on typical data that you encounter to verify that it works in your cases.

Author(s)

Bryan A. Hanson, DePauw University. `<hanson@depauw.edu>`

References

The concept of this function is based on the clever idea in the Stackoverflow post at stackoverflow.com/a/2022348/633251 and relies on code posted at paulbourke.net/geometry/pointlineplane/pointline.r (referenced in the SO post). Minor modifications to the code were made to that code in order to vectorize it.

`getCorrelationRange` *Compute the correlation range values for all genes in the gene-gene correlation matrix.*

Description

Compute the correlation range values for all genes in the gene-gene correlation matrix.

Usage

```
getCorrelationRange(correlation_matrix)
```

Arguments

`correlation_matrix`
gene-gene correlation matrix

Value

list of p-values, adjusted p-values and correlation ranges for each gene

---

`getFilteredData` *Filter the dataset by removing lowly expressed genes and mitochondrial, spike-in and ribosomal genes*

Description

Filter the dataset by removing lowly expressed genes and mitochondrial, spike-in and ribosomal genes

Usage

```
getFilteredData(data, min.cells = 0.05 * ncol(data), species = "human")
```

Arguments

`data`	gene expression matrix
`min.cells`	gene expression matrix
`species`	species to use for gene filtering: "human" (default), "mouse" and "rat"

Value

filtered gene-expression matrix

---

getGGC — *Compute the correlation range values for all genes in the gene-gene correlation matrix*

Description

Compute the correlation range values for all genes in the gene-gene correlation matrix

Usage

```
getGGC(data)
```

Arguments

`data`	log-transformed gene-expression matrix

Value

list of genes with their z-transformed correlation range values

---

getOptimalFeatureSet — *Determine the optimal feature set using Density Index (DI)*

Description

Determine the optimal feature set using Density Index (DI)

Usage

```
getOptimalFeatureSet(
  filt.data,
  ordered.genes,
  elbow.pt = 25,
  k = 10,
  num.pcs = 20,
  error = 0
)
```

Arguments

`filt.data`	log-transformed filtered gene-expression matrix
`ordered.genes`	genes ordered after stepwise regression
`elbow.pt`	Elbow point to start determining optimal feature set
`k`	number of nearest neighbours for CI computation
`num.pcs`	number of principal components to represent sc data. Default is 20.
`error`	Acceptable error margin for kNN computation. Default is 0, but is set to 1 for large datasets.

Value

optimal set of feature genes

`logNormalize` *Log-transform and normalize data by sequencing depth*

Description

Log-transform and normalize data by sequencing depth

Usage

```
logNormalize(raw.data, scale.factor = 10000)
```

Arguments

`raw.data`	raw gene expression matrix
`scale.factor`	scaling factor for normalization

Value

log-normalized gene expression matrix

`pbmc_norm_small_data` *Small PBMC dataset*

Description

Normalized and log-transformed data from the pbmc_small object of the Seurat package.

Usage

```
pbmc_norm_small_data
```

Format

An object of class `dgCMatrix` with 230 rows and 80 columns.

References

Hao et al. (2020). bioRxiv (doi: 10.1101/2020.10.12.335331).

`runStepwiseReg` *Run step-wise regression to order the features*

Description

Run step-wise regression to order the features

Usage

```
runStepwiseReg(ggc)
```

Arguments

`ggc`	gene-gene correlation matrix

Value

optimal feature set

Index