Package ‘ClussCluster’

October 12, 2022

Type Package

Title Simultaneous Detection of Clusters and Cluster-Specific Genes in High-Throughput Transcriptome Data

Version 0.1.0

Description Implements a new method 'ClussCluster' descried in Ge Jiang and Jun Li, "Simultaneous Detection of Clusters and Cluster-Specific Genes in High-throughput Transcriptome Data" (Unpublished).
Simultaneously perform clustering analysis and signature gene selection on high-dimensional transcriptome data sets. To do so, 'ClussCluster' incorporates a Lasso-type regularization penalty term to the objective function of K-means so that cell-type-specific signature genes can be identified while clustering the cells.

Depends R (>= 2.10.0)

Suggests knitr, rmarkdown (>= 1.13)

VignetteBuilder knitr

Imports stats (>= 3.5.0), utils (>= 3.5.0), VennDiagram, scales (>= 1.0.0), reshape2 (>= 1.4.3), ggplot2 (>= 3.1.0), rlang (>= 0.3.4)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Li Jun [cre],
    Jiang Ge [aut],
    Wang Chuanqi [ctb]

Maintainer Li Jun <jun.li@nd.edu>

Repository CRAN

Date/Publication 2019-07-02 16:30:16 UTC
ClussCluster

Description

ClussCluster takes the single-cell transcriptome data and returns an object containing cell types and type-specific signature gene sets.

Selects the tuning parameter in a permutation approach. The tuning parameter controls the L1 bound on w, the feature weights.

Usage

```r
ClussCluster(x, nclust = NULL, centers = NULL, ws = NULL, nepoch.max = 10, theta = NULL, seed = 1, nstart = 20, iter.max = 50, verbose = FALSE)

ClussCluster_Gap(x, nclust = NULL, B = 20, centers = NULL, ws = NULL, nepoch.max = 10, theta = NULL, seed = 1, nstart = 20, iter.max = 50, verbose = FALSE)
```

Arguments

- `x`: An n x p data matrix. There are n cells and p genes.
- `nclust`: Number of clusters desired if the cluster centers are not provided. If both are provided, nclust must equal the number of cluster centers.
- `centers`: A set of initial (distinct) cluster centres if the number of clusters (nclust) is null. If both are provided, the number of cluster centres must equal nclust.
- `ws`: One or multiple candidate tuning parameters to be evaluated and compared. Determines the sparsity of the selected genes. Should be greater than 1.
- `nepoch.max`: The maximum number of epochs. In one epoch, each cell will be evaluated to determine if its label needs to be updated.
filter_gene

theta
- Optional argument. If provided, theta are used as the initial cluster labels of the ClussCluster algorithm; if not, K-means is performed to produce starting cluster labels.

seed
- This seed is used wherever K-means is used.

nstart
- Argument passed to kmeans. It is the number of random sets used in kmeans.

iter.max
- Argument passed to kmeans. The maximum number of iterations allowed.

verbose
- Print the updates inside every epoch? If TRUE, the updates of cluster label and the value of objective function will be printed out.

B
- Number of permutation samples.

Details

Takes the normalized and log transformed number of reads mapped to genes (e.g., log(RPKM+1) or log(TPM+1) where RPKM stands for Reads Per Kilobase of transcript per Million mapped reads and TPM stands for transcripts per million) but NOT centered.

Value

a list containing the optimal tuning parameter, s, group labels of clustering, theta, and type-specific weights of genes, w.

a list containing a vector of candidate tuning parameters, ws, the corresponding values of objective function, O, a matrix of values of objective function for each permuted data and tuning parameter, 0_b, gap statistics and their one standard deviations, Gap and sd.Gap, the result given by ClussCluster, run, the tuning parameters with the largest Gap statistic and within one standard deviation of the largest Gap statistic, bestw and onesd.bestw.

Examples

data(Hou_sim)
hou.dat <- Hou_sim$x
run.ft <- filter_gene(hou.dat)
hou.test <- ClussCluster(run.ft$dat.ft, nclust = 3, ws = 4, verbose = FALSE)

filter_gene

Gene Filter

Description

Filters out genes that are not suitable for differential expression analysis.

Usage

filter_gene(dfname, minmean = 2, n0prop = 0.2, minsd = 1)
Arguments

- **dfname**: name of the expression data frame
- **minmean**: minimum mean expression for each gene
- **n0prop**: minimum proportion of zero expression (count) for each gene
- **minsd**: minimum standard deviation of expression for each gene

Details

Takes an expression data frame that has been properly normalized but NOT centered. It returns a list with the slot `dat.ft` being the data set that satisfies the pre-set thresholds on minimum mean, standard deviation (sd), and proportion of zeros (n0prop) for each gene.

If the data has already been centered, one can still apply the filters of mean and sd but not n0prop.

Value

A list containing the data set with genes satisfying the thresholds, `dat.ft`, the name of `dat.ft`, and the indices of those kept genes, `index`.

Examples

```r
dat <- matrix(rnbinom(300*60, mu = 2, size = 1), 300, 60)
dat_filtered <- filter_gene(dat, minmean=2, n0prop=0.2, minsd=1)
```

Description

This data contains expression levels (normalized and log-transformed) for 33 cells and 100 genes.

Usage

```r
data(Hou_sim)
```

Format

An object containing the following variables:

- `x`: An expression data frame of 33 HCC cells on 100 genes.
- `y`: Numerical group indicator of all cells.
- `gnames`: Gene names of all genes.
- `snames`: Cell names of all cells.
- `groups`: Cell group names.
- `note`: A simple note of the data set.
plot_ClussCluster

Details

This data contains raw expression levels (log-transformed but not centered) for 33 HCC cells and 100 genes. The 33 cells belongs to three different subpopulations and exhibited different biological characteristics. For descriptions of how we generated this data, please refer to the paper.

Source


References


Examples

data(Hou_sim)
data <- Hou_sim$x

plot_ClussCluster(object, m = 10, snames = NULL, gnames = NULL, ...)
top.m.hm(object, m, snames = NULL, gnames = NULL, ...)

Description

Plots the number of signature genes against the tuning parameters if multiple tuning parameters are evaluated in the object. If only one is included, then plot_ClussCluster returns a venn diagram and a heatmap at this particular tuning parameter.

Usage

plot_ClussCluster(object, m = 10, snames = NULL, gnames = NULL, ...)
top.m.hm(object, m, snames = NULL, gnames = NULL, ...)

Arguments

object An object that is obtained by applying the ClussCluster function to the data set.
m The number of top signature genes selected to produce the heatmap.
snames The names of the cells.
gnames The names of the genes
... Additional parameters, sent to the method
Details

Takes the normalized and log transformed number of reads mapped to genes (e.g., \( \log(\text{RPKM}+1) \) or \( \log(\text{TPM}+1) \) where RPKM stands for Reads Per Kilobase of transcript per Million mapped reads and TPM stands for transcripts per million) but NOT centered.

If multiple tuning parameters are evaluated in the object, the number of signature genes is computed for each cluster and is plotted against the tuning parameters. Each color and line type corresponds to a cell type.

If only one tuning parameter is evaluated, two plots will be produced. One is the venn diagram of the cell-type-specific genes, the other is the heatmap of the data with the cells and top \( m \) signature genes. See more details in the paper.

Value

a ggplot2 object of the heatmap with top signature genes selected by ClussCluster

Examples

data(Hou_sim)
run.cc <- ClussCluster(Hou_sim$x, nclust = 3, ws = c(2.4, 5, 8.8))
plot_ClussCluster(run.cc, m = 5, snames=Hou$snames, gnames=Hou$gnames)

Description

Plots the results of ClussCluster_Gap

Usage

plot_ClussCluster_Gap(object)

Arguments

object object obtained from ClussCluster_Gap()
print_ClussCluster

**Description**

Prints out the results of ClussCluster

**Usage**

```
print_ClussCluster(object)
```

**Arguments**

- `object` An object that is obtained by applying the ClussCluster function to the data set.

---

print_ClussCluster_Gap

**Description**

Prints out the results of ClussCluster_Gap. Prints the gap statistics and number of genes selected for each candidate tuning parameter.

**Usage**

```
print_ClussCluster_Gap(object)
```

**Arguments**

- `object` An object that is obtained by applying the ClussCluster_Gap function to the data set.
Description

An example data set containing expressing levels for 60 cells and 200 genes. The 60 cells belong to 4 cell types with 15 cells each. Each cell type is uniquely associated with 30 signature genes, i.e., the first cell type is associated with the first 30 genes, the second cell type is associated with the next 30 genes, so on and so forth. The remaining 80 genes show indistinct expression patterns among the four cell types and are considered as noise genes.

Usage

data(sim_dat)

Format

A data frame with 60 cells on 200 genes.

Value

A simulated dataset used to demonstrate the application of ClussCluster.

Examples

data(sim_dat)
head(sim_dat)
Index

* datasets
  Hou_sim, 4
  sim_dat, 8
ClussCluster, 2
ClussCluster_Gap (ClussCluster), 2
filter_gene, 3
Hou_sim, 4
plot_ClussCluster, 5
plot_ClussCluster_Gap, 6
print_ClussCluster, 7
print_ClussCluster_Gap, 7
sim_dat, 8
top.m.hm (plot_ClussCluster), 5