Package: MBCluster.Seq (via r-universe)

November 15, 2026

Type Package

Title Model-Based Clustering for RNA-Seq Data

Version 1.0.9001

Author Yaqing Si [aut, cre]

Maintainer Yaqing Si <siyaqing@gmail.com>

Description Cluster genes based on Poisson or Negative-Binomial model for RNA-Seq or other digital gene expression (DGE) data. Compatibility updates by Renova.

License GPL (>= 3)

Imports grDevices, graphics, stats, utils

LazyLoad yes

Repository https://renova-r.r-universe.dev

Date/Publication 2026-11-15 20:36:38 UTC

RemoteUrl https://github.com/renova-r/MBCluster.Seq

RemoteRef renova-1.0.9001

RemoteSha 794a5c85267aebadfb3c3715828fa0bae4e262e1

Contents

Cluster.RNASeq 2
Count 3
Hybrid.Tree 4
KmeansPlus.RNASeq 5
MBCluster.Seq.Internal 6
plotHybrid.Tree 6
RNASeq.Data 7

Index **9**

`Cluster.RNASeq` *Do clustering for count data based on poisson or negative-binomial model*

Description

Given a set of initial cluster centers and specify the iteration algorithm, the function proceed the model-based clustering.

Usage

```
Cluster.RNASeq(data, model, centers = NULL, method = c("EM", "DA", "SA"),
  iter.max = 30, TMP = NULL)
```

Arguments

`data`	RNA-seq data from output of function RNASeq.Data()
`model`	Currently could be either Poisson or negative-binomial model for count data
`centers`	Initial cluster centers as a matrix of K rows and I columns to start the clustering algorithm. Each rows is mean-centered to have zero sum. A recommended initial set can be obtained by KmeansPlus.RNASeq()
`method`	Iteration algorithm to update the estimates of cluster and their centers. Could be Expectation-Maximization (EM), Deterministic Annealing (DA) or Simulated Annealing (SA).
`iter.max`	The maximum number of iterations allowed
`TMP`	The 'temperature' serving as annealing rate for DA and SA algorithms. The default setting starts from TMP=4 with decreasing rate 0.9

Value

`probability`	a matrix containing the probability of each gene belonging to each cluster
`centers`	estimates of the cluster centers, a matrix with the same dimension as the initial input
`cluster`	a vector taking values between 1,2,...,K, indicating the assignments of the objects to the clusters

References

Model-Based Clustering for RNA-seq Data, Yaqing Si , Peng Liu, Pinghua Li and Thomas Brutnell

Examples

```
###### run the following codes in order
#
# data("Count")     ## a sample data set with RNA-seq expressions
#                   ## for 1000 genes, 4 treatment and 2 replicates
# head(Count)
# GeneID=1:nrow(Count)
# Normalizer=rep(1,ncol(Count))
# Treatment=rep(1:4,2)
# mydata=RNASeq.Data(Count,Normalize=NULL,Treatment,GeneID)
#                   ## standardized RNA-seq data
# c0=KmeansPlus.RNASeq(mydata,nK=10)$centers
#                   ## choose 10 cluster centers to initialize the clustering
# cls=Cluster.RNASeq(data=mydata,model="nbinom",centers=c0,method="EM")$cluster
#                   ## use EM algorithm to cluster genes
# tr=Hybrid.Tree(data=mydata,cluste=cls,model="nbinom")
#                   ## bulild a tree structure for the resulting 10 clusters
# plotHybrid.Tree(merge=tr,cluster=cls,logFC=mydata$logFC,tree.title=NULL)
#                   ## plot the tree structure
```

`Count` *Sample of Count Data*

Description

The `Count` data frame consists of 1000 genes with 4 treatment groups and 2 biological replicates

Format

This data frame contains 8 columns of count, with colnames as `N1.1` `N2.1` `N3.1` `N4.1` `N1.2` `N2.2` `N3.2` `N4.2`

Examples

```
data("Count")
head(Count)
#     N1.1 N2.1 N3.1 N4.1 N1.2 N2.2 N3.2 N4.2
#[1,]    2    0    0    0    4    0    0    0
#[2,]    4  357 2537 1295   19 1056 2690 4411
#[3,]    0    0    6    8    1    2    8   18
#[4,]    1    1    1    0    2    5    1    2
#[5,]    2   10  107   32    2   31   94   69
#[6,]   79    8   18    5  102   24   21   14
```

`Hybrid.Tree` *Do hybrid-hierarchical clustering for RNA-seq data*

Description

The hybrid-hierarchical clustering starts from an initial partition of the objects, and merges the small clusters gradually into one tree structure

Usage

```
Hybrid.Tree(data, cluster0, model = "nbinom")
```

Arguments

`data`	RAN-seq data standardized by RNASeq.Data()
`cluster0`	A partition of the objects, should be a vector with values ranging from 1 to K0, where K0 is the number of small clusters at the bottom of the hierarchical structure.
`model`	The probability models to calculated the distance between to merged clusters

Value

a table is returned to keep the information of the tree structure. The table has K rows and 2 columns, where K is the maximum level of the tree, and each row shows the two node being merged in each step

Examples

```
###### run the following codes in order
#
# data("Count")     ## a sample data set with RNA-seq expressions
#                   ## for 1000 genes, 4 treatment and 2 replicates
# head(Count)
# GeneID=1:nrow(Count)
# Normalizer=rep(1,ncol(Count))
# Treatment=rep(1:4,2)
# mydata=RNASeq.Data(Count,Normalize=NULL,Treatment,GeneID)
#                   ## standardized RNA-seq data
# c0=KmeansPlus.RNASeq(mydata,nK=10)$centers
#                   ## choose 10 cluster centers to initialize the clustering
# cls=Cluster.RNASeq(data=mydata,model="nbinom",centers=c0,method="EM")$cluster
#                   ## use EM algorithm to cluster genes
# tr=Hybrid.Tree(data=mydata,cluste=cls,model="nbinom")
#                   ## bulild a tree structure for the resulting 10 clusters
# plotHybrid.Tree(merge=tr,cluster=cls,logFC=mydata$logFC,tree.title=NULL)
#                   ## plot the tree structure
```

KmeansPlus.RNASeq *Initialize the cluster centroids by a model-based Kmeans++ algorithm*

Description

The cluster centroids are initialized by a method analogy to Arthur and Vassilvitskii (2007)'s Kmeans++ algorithm

Usage

```
KmeansPlus.RNASeq(data, nK, model ="nbinom", print.steps=FALSE)
```

Arguments

data	RNA-Seq data from output of function RNASeq.Data()
nK	The preselected number of cluster centroids
model	The probability model for the count data. The distances between the cluster centroids will be calculated based on the likelihood functions. The model can be 'poisson' for Poisson or 'nbinom' for negative binomial distribution.
print.steps	print out the proceeding steps or not

Value

centers	a matrix of nK rows which contains the value cluster centroids. A chosen cluster centroid is the log fold change (log-FC) of a gene across different treatments, normalized to have zero-sum
ID	The ID number of the selected genes whose log-FC are used as the initial cluster centroids

Examples

```
###### run the following codes in order
#
# data("Count")     ## a sample data set with RNA-seq expressions
#                   ## for 1000 genes, 4 treatment and 2 replicates
# head(Count)
# GeneID=1:nrow(Count)
# Normalizer=rep(1,ncol(Count))
# Treatment=rep(1:4,2)
# mydata=RNASeq.Data(Count,Normalize=NULL,Treatment,GeneID)
#                   ## standardized RNA-seq data
# c0=KmeansPlus.RNASeq(mydata,nK=10)$centers
#                   ## choose 10 cluster centers to initialize the clustering
# cls=Cluster.RNASeq(data=mydata,model="nbinom",centers=c0,method="EM")$cluster
#                   ## use EM algorithm to cluster genes
# tr=Hybrid.Tree(data=mydata,cluste=cls,model="nbinom")
#                   ## bulild a tree structure for the resulting 10 clusters
# plotHybrid.Tree(merge=tr,cluster=cls,logFC=mydata$logFC,tree.title=NULL)
#                   ## plot the tree structure
```

MBCluster.Seq.Internal *Internal function for MBCluster.Seq package*

Description

Internal functions for MBCluster.Seq package

plotHybrid.Tree *Plot the tree structure of the hybrid-hierarchical clustering results.*

Description

Each vertical bar at the bottom represents the profile of one genes, with the colors indicating the log folder changes relative to the mean expression of the gene. The number at the bottom shows the labels of the smallest clusters

Usage

```
plotHybrid.Tree(merge, cluster, logFC, tree.title = NULL,colorful=FALSE)
```

Arguments

merge	the merging steps to build the tree, can be the results of Hybrid.Tree()
cluster	The assignment of genes at the bottom of the tree, should be the same as the input for Hybrid.Tree
logFC	The log-fold change of each gene, a table of G rows and I columns
tree.title	The title of the plot
colorful	if FALSE, plot will be in black-white color; if TRUE, plot will be in heat colors (library 'grDevices' might be needed).

Examples

```
###### run the following codes in order
#
# data("Count")     ## a sample data set with RNA-seq expressions
#                   ## for 1000 genes, 4 treatment and 2 replicates
# head(Count)
# GeneID=1:nrow(Count)
# Normalizer=rep(1,ncol(Count))
# Treatment=rep(1:4,2)
# mydata=RNASeq.Data(Count,Normalize=NULL,Treatment,GeneID)
#                   ## standardized RNA-seq data
# c0=KmeansPlus.RNASeq(mydata,nK=10)$centers
#                   ## choose 10 cluster centers to initialize the clustering
```

```
# cls=Cluster.RNASeq(data=mydata,model="nbinom",centers=c0,method="EM")$cluster
#                   ## use EM algorithm to cluster genes
# tr=Hybrid.Tree(data=mydata,cluste=cls,model="nbinom")
#                   ## bulild a tree structure for the resulting 10 clusters
# plotHybrid.Tree(merge=tr,cluster=cls,logFC=mydata$logFC,tree.title=NULL)
#                   ## plot the tree structure
```

`RNASeq.Data` *Standardize RNASeq Data for Clustering*

Description

RNASeq.Data is used to collect RNA-Seq data that need to be clustered.

Usage

```
RNASeq.Data(Count, Normalizer=NULL, Treatment,GeneID=NULL)
```

Arguments

`Count`	a GxP matrix storing the numbers of reads mapped to G genes in P samples. Non-integer values are allowed.
`Normalizer`	a vector of length P or a GxP matrix to normalize the gene expressions. When Normalizer=NULL, we use log(Q2) by default, where Q3 is the 75
`Treatment`	a vector of length P indicating the assignment of treatments for each column of the Count. For example, Treatment=c(1,1,2,2,3,3) means there are 3 treatments with each having 2 replicates
`GeneID`	the ID's of the genes, labeled by 1,2,...,G if not provided

Value

`GeneID`	ID's of genes provided by the user. Default is 1,2,...,G if not provided
`Treatment`	The same as the input, but is sorted in increasing order.
`Count`	The matrix of counts of reads as provided. The columns of the matrix is re-arranged to match the ordered labels of treatment
`Normalizer`	A matrix contains the input normalization factors as provided or from default setting. If the provided value is a vector, then each column of the matrix will have the same value
`logFC`	A matrix contains the log fold change (log-FC) of the normalized genes expressions across all the treatments. Each row of the log-FC matrix is standardized to has zero sum
`Aver.Expr`	the logarithm of the mean gene expression after normalization
`logFC`	a matrix storing the gene profiles, which is defined as the log fold changes relative to the mean gene expression
`NB.Dispersion`	the estimated gene-wise dispersion if assuming NB model

Examples

```
###### run the following codes in order
#
# data("Count")     ## a sample data set with RNA-seq expressions
#                   ## for 1000 genes, 4 treatment and 2 replicates
# head(Count)
# GeneID=1:nrow(Count)
# Normalizer=rep(1,ncol(Count))
# Treatment=rep(1:4,2)
# mydata=RNASeq.Data(Count,Normalize=NULL,Treatment,GeneID)
#                   ## standardized RNA-seq data
# c0=KmeansPlus.RNASeq(mydata,nK=10)$centers
#                   ## choose 10 cluster centers to initialize the clustering
# cls=Cluster.RNASeq(data=mydata,model="nbinom",centers=c0,method="EM")$cluster
#                   ## use EM algorithm to cluster genes
# tr=Hybrid.Tree(data=mydata,cluste=cls,model="nbinom")
#                   ## bulild a tree structure for the resulting 10 clusters
# plotHybrid.Tree(merge=tr,cluster=cls,logFC=mydata$logFC,tree.title=NULL)
#                   ## plot the tree structure
```

Index

cl.mb *(MBCluster.Seq.Internal)*, 6
cl.nb.est.c *(MBCluster.Seq.Internal)*, 6
cl.nb.est.m *(MBCluster.Seq.Internal)*, 6
cl.nb.est.mc *(MBCluster.Seq.Internal)*, 6
cl.ps.est.mc *(MBCluster.Seq.Internal)*, 6
Cluster.RNASeq, 2
Count, 3

dst.euclidean.pair
(MBCluster.Seq.Internal), 6
dst.KL *(MBCluster.Seq.Internal)*, 6
dst.maximum.pair
(MBCluster.Seq.Internal), 6
dst.pairs *(MBCluster.Seq.Internal)*, 6
dst.pearson.pair
(MBCluster.Seq.Internal), 6
dst.Ward *(MBCluster.Seq.Internal)*, 6
dst2center.pairs
(MBCluster.Seq.Internal), 6

est.nb.mu.mle.one
(MBCluster.Seq.Internal), 6
est.nb.v.QL *(MBCluster.Seq.Internal)*, 6

Hybrid.Tree, 4
Hybrid.Tree.Microarray
(MBCluster.Seq.Internal), 6

in.tree *(MBCluster.Seq.Internal)*, 6

KmeansPlus.RNASeq, 5

leaf.color *(MBCluster.Seq.Internal)*, 6
lglk.cluster *(MBCluster.Seq.Internal)*, 6
lglk.nb *(MBCluster.Seq.Internal)*, 6
lglk.ps *(MBCluster.Seq.Internal)*, 6
loc.node *(MBCluster.Seq.Internal)*, 6

maxRow *(MBCluster.Seq.Internal)*, 6
MBCluster.Seq.Internal, 6
meanCol *(MBCluster.Seq.Internal)*, 6
meanRow *(MBCluster.Seq.Internal)*, 6
MI.1 *(MBCluster.Seq.Internal)*, 6
MI.2 *(MBCluster.Seq.Internal)*, 6
MI.Cluster.Annotation
(MBCluster.Seq.Internal), 6
MI.score *(MBCluster.Seq.Internal)*, 6
minRow *(MBCluster.Seq.Internal)*, 6
move.two *(MBCluster.Seq.Internal)*, 6

NMI.score *(MBCluster.Seq.Internal)*, 6

plotbr *(MBCluster.Seq.Internal)*, 6
plotHybrid.Tree, 6

RNASeq.Data, 7

sortNode *(MBCluster.Seq.Internal)*, 6
sumCol *(MBCluster.Seq.Internal)*, 6
sumRow *(MBCluster.Seq.Internal)*, 6

tree.K *(MBCluster.Seq.Internal)*, 6