Package 'PMixClus'

October 23, 2014

Type Package

Title Penalized Modoel-Based Clustering for RNA-Seq count data

Version 1.0

Date 2014-10-21

Author Ye Tian

Maintainer Ye Tian <tian0049@e.ntu.edu.sgt>

Description Penalized NB or Poisson mixuture model with application to cluster RNA-Seq count data

Suggests MBCluster.Seq

License GPL- (>=3)

LazyData true

R topics documented:

`Cutree` *Cut the Hybrid-Hierarchical tree.*

Description

Cut the HH tree at specific level to get the relative cluster labels.

Usage

```
Cutree(tree, group, level)
```

Arguments

tree	The HH tree structure. It should be the value from HH.Tree.
group	The cluster labels for bottom chrildren. It should be the same with arguments group in HH.Tree.
level	The targetted level. It cannot be larger than the height of the tree.

Value

The cluster labels at the targetted level of HH tree.

Examples

```
# data("seq_count")
# seq_data = PM.pretreat(count = seq_count)
# out.NB = PMixClus(x, 1:3, exp(seq(0, 3, length = 5)))
# tree = HH.Tree(seq_data, out.NB$gt.hat[[3]], out.NB$phi.hat[3, ], out.NB$group[3, ])
# Cutree(tree, out.NB$group[3, ], 2)
```

HH.Tree *Construct the Hybrid-Hierarchical tree.*

Description

Construct the HH tree with the bottom children from penalized model-based method.

Usage

```
HH.Tree(data, gt, phi, group, model = "NB", disp.domain = c(1e-06, 4))
```

Arguments

data	The well prepared data from PM.pretreat
gt	The estimated parameters from PMixClus, e.g. gt[[1]] is the parameters from PMixClus when k = k.init[1].
phi	The estimated dispersion parameters.
group	The bottom group labels for HH.
model	Select the model between negative binomial and Poisson. If choose Poisson, model = "Pois".
disp.domain	Same with disp.domail in PMixClus.

Value

A $k \times 3$ matrix is returned, where k is the number of clusters in the bottom. The first two columns in each row show the two clusters merged in that step and the third column shows the distance of these merged two clusters.

References

Si, Y. and Liu, P. and Li, P. and Thomas, P. B. (2014) Model-based clustering for RNA-seq data. *Bioinformatics*, 30, 197-205

Examples

```
# data("seq_count")
# seq_data = PM.pretreat(count = seq_count)
# out.NB = PMixClus(x, 1:3, exp(seq(0, 3, length = 5)))
# op = which.min(out.NB$BIC)       #Find the minimal BIC or EBIC.
# tree = HH.Tree(seq_data, out.NB$gt.hat[[op]], out.NB$phi.hat[op, ], out.NB$group[op, ])
```

`plotHH.tree` *Plot the Hybrid-Hierarchical tree.*

Description

Plot the Hybrid-Hierarchical tree to visualize the clustering structure. The bottom children are samples that are clustered by penalized model-based method. The samples with same color are in the the same cluster in reality if the true clusters are know in advance. The sample names are shown under the bottom bars.

Usage

```
plotHH.tree(sample.names, tree, group, group.true = NULL, cex.leaf = 0.8,
  tree.title = "Hybrid-Hierarchical Tree")
```

Arguments

`sample.names`	The sample names. It should be consistent with sample names outputted by `PM.pretreat`.
`tree`	The HH tree stucture. It should be the value from `HH.Tree`.
`group`	The group labels for bottom chrildren. It should be the same with arguments `group` in `HH.Tree`.
`group.true`	The true group labels if they are known in advance.
`cex.leaf`	The font size of sample names.
`tree.title`	The title for this figure.

References

Si, Y. and Liu, P. and Li, P. and Thomas, P. B. (2014) Model-based clustering for RNA-seq data. *Bioinformatics*, 30, 197-205

Examples

```
# data("seq_count")
# seq_data = PM.pretreat(count = seq_count)
# out.NB = PMixClus(x, 1:3, exp(seq(0, 3, length = 5)))
# op = which.min(out.NB$BIC)       #Find the minimal BIC or EBIC.
# tree = HH.Tree(seq_data, out.NB$gt.hat[[op]], out.NB$phi.hat[op, ], out.NB$group[op, ])
# group.true = c(rep(1, 5), rep(2, 5))
# plotHH.tree(rownames(data$count), tree, out.NB$group[op, ], group.true = group.true)
```

PM.pretreat *Filter the data and compute the size factor for each sample by median ratio.*

Description

This function can filter out genes that have very low expression and estimate the size factor for each sample by median ratio method. The output data is used in most of other functions in this package.

Usage

```
PM.pretreat(count, csum = 5, sizefactor = NULL)
```

Arguments

count	The $N \times P$ RNA-Seq count matrix with N samples and P genes.
csum	The threshold value to filter out the low expressed genes. The function will remove the genes, the sum of whose read counts over all samples are less than csum.
sizefactor	The size factor for each sample. When sizefactor = NULL, the function will estimate the size factors by median ratio method.

Value

count	The filtered data with row names and column names.
sizefactor	The size factors estimated by this function or inputted by users.

References

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, 11, R106

Examples

```
# data("seq_count")
# PM.pretreat(count = seq_count)
```

PMixClus *Penalized Poisson or NB model based clustering for count data.*

Description

Given the prepared data from PM.pretreat, this function applied penalized Poisson or NB mixture model to do the clustering. The model selection criterions modified BIC and modified EBIC are provided.

Usage

```
PMixClus(data, k.init, lambda, model = "NB", s.update = FALSE,
  MAX_iter = 100, threshold = 1e-07, disp.domain = c(1e-06, 4),
  is.BIC = "BIC")
```

Arguments

data	The well prepared data from PM.pretreat.
k.init	The vector of the number of clusters that the user wants to select from, e.g. k.init = 1:3.
lambda	The vector of tuning parameter that the user wants to select from, e.g. lambda = exp(seq(0, 3, len
model	The distribution used inside the function. By default, the function applys the NB distribution. If model = "Pois", then the Poisson distribution will be applied.
s.update	Whether to update size factors inside the EM for Poisson model.
MAX_iter	The maximal number of iteration for EM.
threshold	The stop threshold for EM algorithm.
disp.domain	The domain in which the function searches for MLE of dispersion parameters. The default value is suitable for most cases in application.
is.BIC	The model selection criterion applied in the function. By defult, is.BIC = "BIC", modified BIC is applied; if is.BIC = "EBIC", the modified EBIC is applied.

Value

BIC	The value of modified BIC or modified EBIC corresponding to k.init.
lambda.sel	The selected tuning parameters by modified BIC or modified EBIC.
theta.cdiff	The matrix to store fluctuation of genes over different clusters. The rows of this matrix correspond to k.init vector respectively and the columns represent genes. In each row the elements are ordered from higher values to lower values.
gnames.st	The sorted gene names according to theta.cdiff. This matrix store the gene names for every element in the theta.cdiff matrix.
group	The group labels for samples. Each row includes the group labels when input corresponding k in k.init.
sizefactor	The size factors. The rows of this matrix represent results from inputting the corresponding elements in k.init. By default, this matrix is fixed on data$sizefactor from PM.pretreat; if model = "Pois" and s.update = T, the size factors are estimated in the EM.
gt.hat	The list contains estimated mean/size factor. This is the necessary argument in HH.Tree.
phi.hat	The estimates of dispersion parameters if model = "NB". This is the necessary argument in HH.Tree.

References

Pan, W. and Shen, X. (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 8, 1145-1164.

Examples

```
# data("seq_count")
# seq_data = PM.pretreat(count = seq_count)
# out.NB = PMixClus(seq_data, 1:3, exp(seq(0, 3, length = 5)))
# out.pois = PMixClus(seq_data, 1:3, exp(seq(0, 3, length = 5)), model = "Pois", s.update=TRUE)
# op = which.min(out.NB$BIC)       #Find the minimal BIC or EBIC.
# k.init[op]                       #The selected number of clusters by BIC or EBIC.
# sum(out.NB$theta.cidff[op, ] == 0)    #The number of genes excluded when \code{k = k.init[op]}.
# out.NB$group[op, ]               #The group labels when \code{k = k.init[op]}.
```

`seq_count` *Simulated RNA-Seq count data*

Description

The data set consist of 1000 genes with 5 biological replicates in each of 2 treatments.

Usage

```
seq_count
```

Format

This data is 10×1000 matrix, in which first 5 samples in the same treatment and the other 5 samples in the other treatment.

Index