Package: clusterSeq (via r-universe)

November 1, 2026

Type Package

Title Clustering of high-throughput sequencing data by identifying co-expression patterns

Version 1.37.0

Depends R (>= 3.0.0), methods, BiocParallel, baySeq, graphics, stats, utils

Imports BiocGenerics

Suggests BiocStyle

Date 2016-01-19

Description Identification of clusters of co-expressed genes based on their expression across multiple (replicated) biological samples.

License GPL-3

LazyLoad yes

biocViews Sequencing, DifferentialExpression, MultipleComparison, Clustering, GeneExpression

URL https://github.com/samgg/clusterSeq

BugReports https://github.com/samgg/clusterSeq/issues

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

Date/Publication 2026-04-28 12:44:49 UTC

RemoteUrl https://github.com/bioc/clusterSeq

RemoteRef HEAD

RemoteSha 39dc31c1a6b1f972eee95418cb1166bd8c44aea4

Contents

clusterSeq-package 2
associatePosteriors 3
cD.ratThymus 4

kCluster 5
makeClusters 7
makeClustersFF 8
plotCluster 9
ratThymus 10
wallace 11

Index **13**

`clusterSeq-package` *Clustering of high-throughput sequencing data by identifying co-expression patterns*

Description

Identification of clusters of co-expressed genes based on their expression across multiple (replicated) biological samples.

Details

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

Index: This package was not yet installed at build time.

Author(s)

Thomas J. Hardcastle [aut], Irene Papatheodorou [aut], Samuel Granjeaud [cre] (ORCID: <https://orcid.org/0000-0001-9245-1535>)

Maintainer: Samuel Granjeaud <samuel.granjeaud@inserm.fr>

Examples

```
#Load in the processed data of observed read counts at each gene for each sample.
data(ratThymus, package = "clusterSeq")

# Library scaling factors are acquired here using the getLibsizes
# function from the baySeq package.
libsizes <- getLibsizes(data = ratThymus)

# Adjust the data to remove zeros and rescale by the library scaling
# factors. Convert to log scale.
ratThymus[ratThymus == 0] <- 1
normRT <- log2(t(t(ratThymus / libsizes)) * mean(libsizes))

# run kCluster on reduced set.
normRT <- normRT[1:1000,]
kClust <- kCluster(normRT)
```

```
# make the clusters from these data.
mkClust <- makeClusters(kClust, normRT, threshold = 1)

# or using likelihood data from a Bayesian analysis of the data

# load in analysed countData object
data(cD.ratThymus, package = "clusterSeq")

# estimate likelihoods of dissimilarity on reduced set
aM <- associatePosteriors(cD.ratThymus[1:1000,])

# make clusters from dissimilarity data
sX <- makeClusters(aM, cD.ratThymus, threshold = 0.5)

# plot first six clusters
par(mfrow = c(2,3))
plotCluster(sX[1:6], cD.ratThymus)
```

`associatePosteriors` *Associates posterior likelihood to generate co-expression dissimilarities between genes*

Description

This function aims to find pairwise dissimilarities between genes. It does this by comparing the posterior likelihoods of patterns of differential expression for each gene, and estimating the likelihood that the two genes are not equivalently expressed.

Usage

```
associatePosteriors(cD, maxsize = 250000, matrixFile = NULL)
```

Arguments

`cD`	A `countData` object containing posterior likelihoods of differential expression for each gene.
`maxsize`	The maximum size (in MB) to use when partitioning the data.
`matrixFile`	If given, a file to write the complete (gzipped) matrix of pairwise distances between genes. Defaults to NULL.

Details

In comparing two genes, we find all patterns of expression considered in the '@groups' slot of the 'cD' (`countData`) object for which the expression of the two genes can be considered monotonic. We then subtract the sum the posterior likelihods of these patterns of expression from 1 to define a likelihood of dissimilarity between the two genes.

Value

A data.frame which for each gene defines its nearest neighbour of higher row index and the dissimilarity with that neighbour.

Author(s)

Thomas J. Hardcastle

See Also

makeClusters makeClustersFF kCluster

Examples

```
# load in analysed countData (baySeq package) object
library(baySeq)
data(cD.ratThymus, package = "clusterSeq")

# estimate likelihoods of dissimilarity on reduced set
aM <- associatePosteriors(cD.ratThymus[1:1000,])
```

---

cD.ratThymus — *Data from female rat thymus tissue taken from the Rat BodyMap project (Yu et al, 2014) and processed by baySeq.*

Description

This data set is a countData object for 17230 genes from 16 samples of female rat thymus tissue. The tissues are extracted from four different age groups (2, 6, 21 and 104 week) with four replicates at each age. Posterior likelihoods for the 15 possible patterns of differential expression have been precalculated using the baySeq-package functions.

Usage

```
data(cD.ratThymus)
```

Format

A countData object

Value

A countData object

Source

Illumina sequencing.

References

Yu Y. et al. A rat RNA-Seq transcriptomic BodyMap across 11 organs and 4 developmental stages. Nature Communications (2014)

See Also

ratThymus

---

kCluster *Constructs co-expression dissimilarities from k-means analyses.*

Description

This function aims to find pairwise distances between genes. It does this by constructing k-means clusterings of the observed (log) expression for each gene, and for each pair of genes, finding the maximum value of k for which the centroids of the clusters are monotonic between the genes.

Usage

```
kCluster(cD, maxK = 100, matrixFile = NULL, replicates =
  NULL, algorithm = "Lloyd", B = 1000, sdm = 1)
```

Arguments

cD	A countData object containing the raw count data for each gene, or a matrix containing the logged and normalised values for each gene (rows) and sample (columns).
maxK	The maximum value of k for which k-means clustering will be performed. Defaults to 100.
matrixFile	If given, a file to write the complete (gzipped) matrix of pairwise distances between genes. Defaults to NULL.
replicates	If given, a factor or vector that can be cast to a factor that defines the replicate structure of the data. See Details.
algorithm	The algorithm to be used by the kmeans function.
B	Number of iterations of bootstrapping algorithm used to establish clustering validity
sdm	Thresholding parameter for validity; see Details.

Details

In comparing two genes, we find the maximum value of k for which separate k-means clusterings of the two genes lead to a monotonic relationship between the centroids of the clusters. For this value of k, the maximum difference between expression levels observed within a cluster of either gene is reported as a measure of the dissimilarity between the two genes.

There is a potential issue in that for genes non-differentially expressed across all samples (i.e., the appropriate value of k is 1), there will nevertheless exist clusterings for k > 1. For some arrangements of data, this leads to misattribution of non-differentially expressed genes. We identify these cases by adapting Tibshirani's gap statistic; bootstrapping uniformly distributed data on the same range as the observed data, calculating the dissimilarity score as above, and finding those cases for which the gap between the bootstrapped mean dissimilarity and the observed dissimilarity for k = 1 exceeds that for k = 2 by more than some multiple (sdm) of the standard error of the bootstrapped dissimilarities of k = 2. These cases are forced to be treated as non-differentially expressed by discarding all dissimilarity data for k > 1.

If the replicates vector is given, or if the replicates slot of a countData given as the 'cD' variable is complete, then the k-means clustering will be done on the median of the expression values of each replicate group. Dissimilarity calculations will still be made on the full data.

Value

A data.frame which for each gene defines its nearest neighbour of higher row index and the dissimilarity with that neighbour.

Author(s)

Thomas J. Hardcastle

See Also

makeClusters makeClustersFF associatePosteriors

Examples

```
#Load in the processed data of observed read counts at each gene for each sample.
data(ratThymus, package = "clusterSeq")

# Library scaling factors are acquired here using the getLibsizes
# function from the baySeq package.
libsizes <- getLibsizes(data = ratThymus)

# Adjust the data to remove zeros and rescale by the library scaling
# factors. Convert to log scale.
ratThymus[ratThymus == 0] <- 1
normRT <- log2(t(t(ratThymus / libsizes)) * mean(libsizes))

# run kCluster on reduced set.
normRT <- normRT[1:1000,]
kClust <- kCluster(normRT)
head(kClust)

# Alternatively, run on a count data object:
# load in analysed countData (baySeq package) object
library(baySeq)
data(cD.ratThymus, package = "clusterSeq")

# estimate likelihoods of dissimilarity on reduced set
```

```
kClust2 <- kCluster(cD.ratThymus[1:1000,])
head(kClust2)
```

`makeClusters` *Creates clusters from a co-expression minimal linkage data.frame.*

Description

This function uses minimal linkage data to perform rapid clustering by singleton agglomeration (i.e., a gene will always cluster with its nearest neighbours provided the distance to those neighbours does not exceed some threshold). For alternative (but slower) clustering options, see the `makeClustersFF` function.

Usage

```
makeClusters(aM, cD, threshold = 0.5)
```

Arguments

`aM`	A data frame constructed by `associatePosteriors` or `kCluster`, defining for each gene the nearest neighbour of higher row index and the dissimilarity with that neighbour.
`cD`	The data given as input to `associatePosteriors` or `kCluster` that produced 'aM'.
`threshold`	A threshold on the maximum dissimilarity at which two genes can cluster. Defaults to 0.5.

Value

An IntegerList object, each member of whom defines a cluster of co-expressed genes. The object is ordered decreasingly by the size of each cluster.

Author(s)

Thomas J Hardcastle

See Also

`makeClustersFF` `kCluster` `associatePosteriors`

Examples

```
#Load in the processed data of observed read counts at each gene for each sample.
data(ratThymus, package = "clusterSeq")

# Library scaling factors are acquired here using the getLibsizes
# function from the baySeq package.
libsizes <- getLibsizes(data = ratThymus)

# Adjust the data to remove zeros and rescale by the library scaling
# factors. Convert to log scale.
ratThymus[ratThymus == 0] <- 1
normRT <- log2(t(t(ratThymus / libsizes)) * mean(libsizes))

# run kCluster on reduced set.
normRT <- normRT[1:1000,]
kClust <- kCluster(normRT)

# make the clusters from these data.
mkClust <- makeClusters(kClust, normRT, threshold = 1)
```

`makeClustersFF` *Creates clusters from a file containing a full dissimilarity matrix.*

Description

This function uses the complete pairwise dissimilarity scores to construct a hierarchical clustering of the genes.

Usage

```
makeClustersFF(file, method = "complete", cut.height = 5)
```

Arguments

`file`	Filename containing the dissimilarity data.
`method`	Method to use in `hclust`.
`cut.height`	Cut height to use in `hclust`.

Value

An IntegerList object containing the clusters derived from a cut hierarchical clustering.

Author(s)

Thomas J Hardcastle

See Also

makeClusters kCluster associatePosteriors

Examples

```
#Load in the processed data of observed read counts at each gene for each sample.
data(ratThymus, package = "clusterSeq")

# Library scaling factors are acquired here using the getLibsizes
# function from the baySeq package.
libsizes <- getLibsizes(data = ratThymus)

# Adjust the data to remove zeros and rescale by the library scaling
# factors. Convert to log scale.
ratThymus[ratThymus == 0] <- 1
normRT <- log2(t(t(ratThymus / libsizes)) * mean(libsizes))

# run kCluster on reduced set. For speed, one thousand bootstraps are
# used, but higher values should be used in real analyses.
# Write full dissimilarity matrix to file "kclust.gz"
normRT <- normRT[1:1000,]
kClust <- kCluster(normRT, B = 1000, matrixFile = "kclust.gz")


# make the clusters from these data.
mkClustR <- makeClustersFF("kclust.gz")

# no need to clean up (specific to Bioconductor pipeline)
file.remove("kclust.gz")
```

---

plotCluster *Plots data from clusterings.*

Description

Given clusterings and expression data, plots representative expression data for each clustering.

Usage

```
plotCluster(cluster, cD, sampleSize = 1000)
```

Arguments

cluster	A list object defining the clusters, produced by makeClusters or makeClustersFF.
cD	The data object used to produce the clusters.
sampleSize	The maximum number of genes that will be ploted.

Details

Expression data are normalised and rescaled before plotting.

Value

Plotting function.

Author(s)

Thomas J Hardcastle

See Also

makeClusters makeClustersFF

Examples

```
# load in analysed countData object
data(cD.ratThymus, package = "clusterSeq")

# estimate likelihoods of dissimilarity on reduced set
aM <- associatePosteriors(cD.ratThymus[1:1000,])

# make clusters from dissimilarity data
sX <- makeClusters(aM, cD.ratThymus, threshold = 0.5)

# plot first six clusters
par(mfrow = c(2,3))
plotCluster(sX[1:6], cD.ratThymus)
```

---

ratThymus *Data from female rat thymus tissue taken from the Rat BodyMap project (Yu et al, 2014).*

Description

This data set is a matrix ('mobData') of raw count data acquired for 17230 genes from 16 samples of female rat thymus tissue. The tissues are extracted from four different age groups (2, 6, 21 and 104 week) with four replicates at each age. Gene annotation is given in the rownames of the matrix.

Usage

```
data(ratThymus)
```

Format

A matrix of RNA-Seq counts in which each of the sixteen columns represents a sample, and each row a gene locus.

Value

A matrix

Source

Illumina sequencing.

References

Yu Y. et al. A rat RNA-Seq transcriptomic BodyMap across 11 organs and 4 developmental stages. Nature Communications (2014)

See Also

cD.ratThymus

---

wallace *Computes Wallace scores comparing two clustering methods.*

Description

Given two clusterings A and B we can calculate the likelihood that two elements are in the same cluster in B given that they are in the same cluster in A, and vice versa.

Usage

```
wallace(v1, v2)
```

Arguments

v1	SimpleIntegerList object (output from makeClusters or makeClustersFF).
v2	SimpleIntegerList object (output from makeClusters or makeClustersFF).

Value

Vector of length 2 giving conditional likelihoods.

Author(s)

Thomas J. Hardcastle

Examples

```
# using likelihood data from a Bayesian analysis of the data

# load in analysed countData object
data(cD.ratThymus, package = "clusterSeq")

# estimate likelihoods of dissimilarity on reduced set
aM <- associatePosteriors(cD.ratThymus[1:1000,])

# make clusters from dissimilarity data
sX <- makeClusters(aM, cD.ratThymus[1:1000,], threshold = 0.5)

# or using k-means clustering on raw count data

#Load in the processed data of observed read counts at each gene for each sample.
data(ratThymus, package = "clusterSeq")

# Library scaling factors are acquired here using the getLibsizes
# function from the baySeq package.
libsizes <- getLibsizes(data = ratThymus)

# Adjust the data to remove zeros and rescale by the library scaling
# factors. Convert to log scale.
ratThymus[ratThymus == 0] <- 1
normRT <- log2(t(t(ratThymus / libsizes)) * mean(libsizes))

# run kCluster on reduced set.
normRT <- normRT[1:1000,]
kClust <- kCluster(normRT, replicates = cD.ratThymus@replicates)

# make the clusters from these data.
mkClust <- makeClusters(kClust, normRT, threshold = 1)


# compare clusterings
wallace(sX, mkClust)
```

Index

∗ **datasets**
    cD.ratThymus, 4
    ratThymus, 10
∗ **manip**
    associatePosteriors, 3
    kCluster, 5
    makeClusters, 7
    makeClustersFF, 8
    wallace, 11
∗ **package**
    clusterSeq-package, 2
∗ **plot**
    plotCluster, 9

associatePosteriors, 3, *6, 7*, *9*

cD.ratThymus, 4, *11*
clusterSeq *(*clusterSeq-package*)*, 2
clusterSeq-package, 2
countData, *3–6*

hclust, *8*

kCluster, *4*, 5, *7*, *9*

makeClusters, *4*, *6*, 7, *9, 10*
makeClustersFF, *4*, *6, 7*, 8, *9, 10*

plotCluster, 9

ratThymus, *5*, 10

wallace, 11