Package ‘Biopeak’

August 21, 2019

Type Package

Title Identification of Impulse-Like Gene Expression Changes in Short Genomic Series Data

Version 1.0

Author David Lauenstein

Maintainer David Lauenstein <david.lauenstein@gmail.com>

Description Enables the user to systematically identify and visualize impulse-like gene expression changes within short genomic series experiments. In order to detect such activation peaks, the gene expression is treated as a signal that propagates along an experimental axis (time, temperature or other series conditions). Peaks are selected by exhaustive identification of local maximums and subsequent filtering based on a range of controllable parameters. Moreover, the 'Biopeak' package provides a series of data exploration tools including: expression profile plots, correlation heat maps and clustering functionalities.

License GPL (>= 2)

Depends R (>= 2.10)

Suggests knitr, rmarkdown

VignetteBuilder knitr

Imports cluster, dbscan, factoextra, gplots, RColorBrewer, stats, graphics

LazyData True

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2019-08-21 09:40:06 UTC

R topics documented:

`bgCorr` *Background noise correction of expression data*

Description

This helper function performs a background noise correction before subjecting the corrected matrix to the peakDetection function. Genes with an expression lower than the 5

Usage

```
bgCorr(exprmat)
```

Arguments

`exprmat`	A numeric matrix with time-series expression data with variables as rownames.

Value

Returns background noise corrected expression matrix.

Author(s)

David Lauenstein

Examples

```
# Example based on the heat-shock dataset
data(heat)
heat = as.matrix(heat)
# Execute the bgCorr function
exprmat_corrected <- bgCorr(heat)
```

`findClusters` *Identification of clusters with similar temporal regulation*

Description

The findClusters function estimates the number of genes with similar temporal regulation and supports three different clustering algorithms: kmeans, dbscan and hierarchical clustering. Clustering is based on a PCA projection of the input data.

Usage

```
findClusters(peakdet, exprmat, maxclusters = 3, eps = 0.02,
  clusters = 3, method = "kmeans")
```

Arguments

`peakdet` A list returned by the peakDetection function.

`exprmat` A numeric matrix with expression series data with variables as rownames.

`maxclusters` Maximal number of clusters used for kmeans cluster estimation.

`eps` Epsilon value used by the dbscan algorithm.

`clusters` Number of clusters used for the cutree function of the hierarchical clustering.

`method` A character string defining the clustering algorithm with options: c('kmeans', 'dbscan', 'hclust').

Value

Returns a cluster assignment of each variable and the number of identified clusters.

Author(s)

David Lauenstein

Examples

```
# Example based on the heat-shock dataset
data(heat)
heat = as.matrix(heat)
# Define series
series <- c(37,40,41,42,43)
# Run the peak detection algorithm
peakdet <- peakDetection(heat, series, type ='rnaseq', actstrength = 1.5,
prominence = 1.3, minexpr = 5000)
# Cluster exploration using kmeans with a maximum of 4 clusters to be assigned
clusters <- findClusters(peakdet, heat, maxclusters = 4, method = 'kmeans')
```

findPeaks *Identification of peaks in an expression signal*

Description

This helper function identifies peaks in an expression signal by treating the gene expression as a signal that propagates along an experimental axis. A peak is defined as a local maximum in the expression signal satisfying: y(t) > y(t+1) and y(t) > y(t-1), where y(t) represents the gene expression as a function of series condition t.

Usage

```
findPeaks(expr)
```

Arguments

expr	A numeric vector with gene expression values

Value

Returns a list comprising of a numeric vector with the location of each peak (peakloc), a numeric vector with the absolute height of each peak (peakheight) and a character vector of gene symbols for which at least one peak has been identified (peakgenes).

Author(s)

David Lauenstein

Examples

```
# Example based on the heat-shock dataset
data(heat)
heat = as.matrix(heat)
# Run the findPeaks function for the first gene in the expression matrix
peaks <- findPeaks(heat[1,])
```

getCormat *Identification of co-expressing genes*

Description

The getCormat function calculates a pair-wise correlation matrix and plots a bi-clustered heatmap.

Usage

```
getCormat(peakdet, exprmat, method = "spearman")
```

Arguments

`peakdet` A list returned by the peakDetection function.

`exprmat` A numeric matrix with expression series data with variables as rownames.

`method` A character string defining the correlation algorithm. Options are: c('pearson', 'kendall', 'spearman').

Value

Returns both the heatmap object and the re-ordered correlation matrix:

Author(s)

David Lauenstein

Examples

```
# Example based on the heat-shock dataset
data(heat)
heat = as.matrix(heat)
# Define series
series <- c(37,40,41,42,43)
# Run the peak detection algorithm
peakdet <- peakDetection(heat, series, type ='rnaseq', actstrength = 1.5,
prominence = 1.3, minexpr = 5000)
# calculate and plot correlation matrix
corobjects <- getCormat(peakdet, heat, method = 'spearman')
```

`heat` *Transcriptional profiles of human epithelial cells in response to heat*

Description

In vitro cultured HEp2 cells were heated at 37 to 43 degrees Celsisu for 60 min and microarray gene expression profiles were acquired at 37, 40, 41, 42 and 43 degrees Celsius.

Usage

```
data(heat)
```

Format

A data frame with 1393 rows and 5 variables:

Hep2_37 gene expression of HEp2 cells at 37 degrees Celsius

Hep2_40 gene expression of HEp2 cells at 40 degrees Celsius

Hep2_41 gene expression of HEp2 cells at 41 degrees Celsius

Hep2_42 gene expression of HEp2 cells at 42 degrees Celsius

Hep2_43 gene expression of HEp2 cells at 43 degrees Celsius

Source

J. M. Laramie, T. P. Chung, B. Brownstein, G. D. Stormo, J. P. Cobb, Transcriptional profiles of human epithelial cells in response to heat: computational evidence for novel heat shock proteins. Shock 29, 623-630 (2008).

maProcessing *Pre-processing of microarray datasets*

Description

This helper function pre-processes microarray datasets by performing an exponentiation with number 2 as the base on the expression values.

Usage

```
maProcessing(expr, exprmat)
```

Arguments

`expr`	A numeric vector with gene expression values
`exprmat`	A numeric matrix with expression series data with variables as rownames.

Value

Returns a numeric vector with the exponentiated expression values.

Author(s)

David Lauenstein

Examples

```
# Example based on the heat-shock dataset
data(heat)
heat = as.matrix(heat)
# Run the findPeaks function for the first gene in the expression matrix
peaks <- maProcessing(heat[1,], heat)
```

peakDetection *Identification of biomarkers specific to distinct phases of the underlying biological process*

Description

The peakDetection function facilitates the identification of impulse-like gene expression changes based on user-defined selection criteria. This function calls the helper functions: bgCorr(), maProcessing() and findPeaks().

Usage

```
peakDetection(exprmat, series, actstrength = 1.3, prominence = 1.3,
  type = "rnaseq", minexpr = 0, peakwidth = 0, sustact = 0.6,
  bgcorr = T)
```

Arguments

`exprmat`	A numeric matrix with expression series data with variables as rownames.
`series`	A numeric vector defining the experimental series (e.g. time-points of sample acquisition).
`actstrength`	Threshold for minimal activation relative to the mean expression across all time-points.
`prominence`	Threshold for minimal peak prominence relative to the second highest peak.
`type`	A character string defining the sequencing platform. Possible values are c('microarray', 'rnaseq').
`minexpr`	An optional threshold for minimal mean expression across all time-points for a given gene.
`peakwidth`	An optional definitino of the minimal number of time-points that a peak spans (based on sustact threshold).
`sustact`	An optional threshold for minimal peakheight relative to the main peak to be considered as sustained activation.
`bgcorr`	An optional logical constant (TRUE or FALSE) defining if a background noise correction is performed or not.

Value

Returns a list comprising of multiple vectors and matrices. A numeric vector with the location of each peak (peakloc), a numeric vector with the absolute height of each peak (peakheight), a character vector of gene symbols for which at least one peak has been identified (peakgenes), a numeric matrix containing time-points with sustained activation, the logical vector defining which gene index has been selected and the numeric input vector defining the time-series.

Author(s)

David Lauenstein

Examples

```
# Example based on the heat-shock dataset
data(heat)
heat = as.matrix(heat)
# Define series
series <- c(37,40,41,42,43)
# Run the peak detection algorithm
peakdet <- peakDetection(heat, series, type ='rnaseq', actstrength = 1.5,
prominence = 1.3, minexpr = 5000)
```

`plotExpression` *Plot the expression signal of individual genes*

Description

This function plots the expression signal of a defined gene and marks the main peak location with a dashed line.

Usage

```
plotExpression(exprmat, gene, series, peakdet)
```

Arguments

`exprmat`	A numeric matrix with expression series data with variables as rownames.
`gene`	A character string (not case-sensitive) defining the gene to be plotted.
`series`	A numeric vector defining the experimental series (e.g. time-points of sample acquisition).
`peakdet`	A list returned by the peakDetection function.

Value

This function does not return any value but generates a plot.

Author(s)

David Lauenstein

Examples

```
# Example based on the heat-shock dataset
data(heat)
heat = as.matrix(heat)
# Define series
series <- c(37,40,41,42,43)
# Run the peak detection algorithm
peakdet <- peakDetection(heat, series, type ='rnaseq', actstrength = 1.5,
prominence = 1.3, minexpr = 5000)
# Plot the expression signal of the gene CXCL5
plotExpression(heat, 'CXCL5', series, peakdet)
```

plotHeatmap *Plot a heatmap for selected genes*

Description

This function acts as a wrapper function for the heatmap.2 function of the gplots package and normalizes the subjected expression matrix to the log2 of the mean expression of the gene across all time-points.

Usage

```
plotHeatmap(peakdet, exprmat, clustermembers = c())
```

Arguments

peakdet	A list returned by the peakDetection function.
exprmat	A numeric matrix with expression series data with variables as rownames.
clustermembers	An optional character vector defining genes to be selected.

Value

This function does not return any value but generates a heatmap plot.

Author(s)

David Lauenstein

Examples

```
# Example based on the heat-shock dataset
data(heat)
heat = as.matrix(heat)
# Define series
series <- c(37,40,41,42,43)
# Run the peak detection algorithm
peakdet <- peakDetection(heat, series, type ='rnaseq', actstrength = 1.5,
prominence = 1.3, minexpr = 5000)
# cluster exploration using kmeans with a maximum of 4 clusters to be assigned
clusters <- findClusters(peakdet, heat, maxclusters = 4, method = 'kmeans')
# Plot the heatmap for one of the clusters returned by the findClusters function
heatmap <- plotHeatmap(peakdet, heat, clustermembers = clusters$clustermembers[[1]])
```

`saveOutput` *Save the peak detection output to a text file*

Description

This function saves the output of the peakDetetection funtion (peakgenes, peaklocation and peakheight) to a text file.

Usage

```
saveOutput(peakdet, filename)
```

Arguments

`peakdet`	A list returned by the peakDetection function.
`filename`	A character string defining the output file.

Value

This function does not return any value but saves data to a text file.

Author(s)

David Lauenstein

Index