Package ‘ClueR’

February 19, 2015

Type Package
Title CLUster Evaluation (CLUE)
Version 1.0
Date 2015-2-1
Author Pengyi Yang & Raja Jothi
Maintainer Pengyi Yang <yangpy7@gmail.com>
Description CLUE is an R package for identifying optimal number of clusters in a given time-course dataset clustered by cmeans or kmeans algorithms.
License GPL-3
Depends R (>= 2.10.0), e1071
Suggests Hmisc
NeedsCompilation no
Repository CRAN
Date/Publication 2015-02-05 00:18:20

R topics documented:

ClueR-package .................................................. 2
clustEnrichment ............................................. 2
clustOptimal ................................................... 3
enrichmentTest ................................................. 5
fuzzPlot ......................................................... 6
hES ............................................................. 6
Pathways.KEGG .............................................. 7
Pathways.reactome .......................................... 7
PhosphoSite.human .......................................... 7
PhosphoSite.mouse .......................................... 7
runClue ......................................................... 8
temporalSimu .................................................. 9

Index 11
Description

CLUster Evaluation (or "CLUE") is an R package for identifying optimal number of clusters in a given time-course dataset clustered by cmeans or kmeans algorithms. It relies on a reference annotation set to test for enrichment in each cluster using Fisher’s Exact Test and then test for overall enrichment of the entire clusters using Fisher’s combined probability test.

CLUE is designed for analyzing time-course phosphoproteomics dataset using kinase-substrate annotation as reference. However, it can be applied to time-course microarray dataset as well by replacing the kinase-substrate annotation with gene sets annotation.

Details

Package: CLUE
Type: Package
Version: 1.0
Date: 2015-2-1
License: GPL-3

Author(s)

Pengyi Yang

References

Pengyi Yang, Xiaofeng Zheng, Vivek Jayaswal, Guang Hu, Yee Hwa Yang, Raja Jothi, Detecting key signaling events from time-series phosphoproteomics data, submitted.

description

clostEnrichment

Description

Takes a clustering object generated by cmeans or kmeans algorithm and determine the enrichment of each cluster and then the overall enrichment of this clustering object based on an annotation file.

Usage

clustEnrichment(clustObj, annotation, effectiveSize, pvalueCutoff = 0.05)
**Argument**

- **clustObj**: the clustering object generated by cmeans or kmeans.
- **annotation**: a list with names correspond to kinases and elements correspond to substrates belonging to each kinase.
- **effectiveSize**: the size of kinase-substrate groups to be considered for calculating enrichment. Groups that are too small or too large will be removed from calculating overall enrichment of the clustering.
- **pvalueCutoff**: a p-value cutoff for determining which kinase-substrate groups to be included in calculating overall enrichment of the clustering.

**Value**

A list that contains both the p-value indicating the overall enrichment and a sublist that details the enrichment of each individual cluster.

**Examples**

```r
# simulate a time-series data with six distinctive profile groups and each group with
# a size of 500 phosphorylation sites.
simuData <- temporalsimu(seed=1, grposize=500, sdd=1, numGroups=4)

# create an artificial annotation database. Generate 100 kinase-substrate groups each
# comprising 50 substrates assigned to a kinase.
# among them, create 5 groups each contains phosphorylation sites defined to have the
# same temporal profile.
kinaseAnno <- list()
groupSize <- 500
for (i in 1:5) {
  kinaseAnno[[i]] <- paste("p", (groupSize*(i-1)+1):(groupSize*(i-1)+50), sep="_")
}
for (i in 6:100) {
  set.seed(i)
  kinaseAnno[[i]] <- paste("p", sample.int(nrow(simuData), size=50), sep="_")
}
names(kinaseAnno) <- paste("KS", 1:100, sep="_")

# testing enrichment of clustering results by partition the data into six clusters
# using cmeans algorithm.
library(e1071)
clustObj <- cmeans(simuData, centers=6, iter.max=50, m=1.25)
clustEnrichment(clustObj, annotation=kinaseAnno, effectiveSize=c(5, 100), pvalueCutoff=0.05)
```

---

**clustOptimal**

*Generate optimal clustering*
clustOptimal

Description

Takes a clue output and generate the optimal clustering of the time-course data.

Usage

clustOptimal(clueObj, rep, user.maxK = NULL, visualize = TRUE, ...)

Arguments

clueObj  the output from runClue.
rep  number of times the clustering is to be applied to find the best clustering result.
user.maxK  user defined optimal k value for generating optimal clustering. If not provided, the optimal k that is identified by clue will be used.
visualize  a boolean parameter indicating whether to visualize the clustering results.
...  pass additional parameter for controlling the plot if visualize is TRUE.

Value

return a list containing optimal clustering object and enriched kinases or gene sets.

Examples

# simulate a time-series data with 4 distinctive profile groups and each group with # a size of 50 phosphorylation sites.
simuData <- temporalsimu(seed=1, groupSize=50, sdd=1, numGroups=4)

# create an artificial annotation database. Generate 20 kinase-substrate groups each # comprising 10 substrates assigned to a kinase.
# among them, create 4 groups each contains phosphorylation sites defined to have the # same temporal profile.
kInaseAnno <- list()
groupSize <- 50
for (i in 1:4) {
  kinaseAnno[[i]] <- paste("p", (groupSize*(i-1)+1):(groupSize*(i-1)+10), sep="_")
}
for (i in 5:20) {
  set.seed(i)
  kinaseAnno[[i]] <- paste("p", sample.int(nrow(simuData), size = 10), sep="_")
} names(kInaseAnno) <- paste("KS", 1:20, sep="_")

# run CLUE with a repeat of 2 times and a range from 2 to 7
set.seed(1)
clueObj <- runClue(Tc=simuData, annotation=kInaseAnno, rep=5, kRange=7)

# visualize the evaluation outcome
Ms <- apply(clueObj$evMat, 2, mean, na.rm=TRUE)
Ss <- apply(clueObj$evMat, 2, sd, na.rm=TRUE)
library(Hmisc)
###enrichmentTest

- **Fisher's exact test-based enrichment test**

####Description

Takes a vector of names representing phosphorylation sites that are partitioned in the same cluster and an kinase-substrate annotation. Test for enrichment of the kinase based on the name vector.

####Usage

```r
enrichmentTest(clust, annotation, universe, alter = "greater")
```

####Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clust</td>
<td>a vector of names representing phosphorylation sites that are partitioned in the same cluster</td>
</tr>
<tr>
<td>annotation</td>
<td>a list with names correspond to kinases and elements correspond to substrates belong to each kinase</td>
</tr>
<tr>
<td>universe</td>
<td>the universe of names to compare against</td>
</tr>
<tr>
<td>alter</td>
<td>indicates the alternative hypothesis and must be one of &quot;two.sided&quot;, &quot;greater&quot; or &quot;less&quot;</td>
</tr>
</tbody>
</table>

####Value

a matrix that contains enrichment of each kinase based on the input name vector.
fuzzplot

Visualize fuzzy clustering results

Description

Takes in a time-course matrix and its clustering results as a cmeans clustering object. Produce a plot to visualize the clustering results.

Usage

```r
fuzzplot(Tc, clustObj, mfrow = c(1, 1), cols, min.mem = 0,
         new.window = FALSE, lwd = 3)
```

Arguments

- `Tc`: a numeric matrix to be clustered. The columns correspond to the time-course and the rows correspond to phosphorylation sites.
- `clustObj`: the clustering of Tc generated from cmeans or kmeans clustering.
- `mfrow`: control the subplots in a graphic window.
- `cols`: color palette to be used for plotting. If the color argument remains empty, the default palette is used.
- `min.mem`: phosphorylation sites with membership values below min.mem will not be displayed.
- `new.window`: should a new window be opened for graphics.
- `lwd`: line width. Default is 3.

Examples

```r
# load the human ES phosphoproteomics data (Rigbolt et al. Sci Signal. 4(164):rs3, 2011)
data(hES)
# apply cmeans clustering to partition the data into 11 clusters
library(e1071)
clustObj <- cmeans(hES, centers=11, iter.max=50, m=1.25)
# visualize clustering results
fuzzplot(hES, clustObj, mfrow = c(3,4))
```

hES

Human embryonic stem cell phosphoproteomics data

Description

The data object contains a time-course phosphoproteomics profiling of the human embryonic stem cells through differentiation. The differentiation is induced by using PMA and the time points of 30m, 1hr, 6hrs, and 24hrs are measured and the ratio are taken against 0m. For details please refer to the article: Rigbolt et al. Sci Signal. 4(164):rs3, 2011
Description

The data object contains the annotations of KEGG pathways.

Pathways.reactome Reactome pathway annotations

Description

The data object contains the annotations of reactome pathways.

PhosphoSitePlus annotations for human

Description

The data object contains the annotations of kinases and their corresponding substrates as phosphorylation sites in human. It is extracted from the PhosphoSitePlus database. For details of PhosphoSitePlus, please refer to the article: Hornbeck et al. Nucleic Acids Res. 40:D261-70, 2012

PhosphoSitePlus annotations for mouse

Description

The data object contains the annotations of kinases and their corresponding substrates as phosphorylation sites in mouse. It is extracted from the PhosphoSitePlus database. For details of PhosphoSitePlus, please refer to the article: Hornbeck et al. Nucleic Acids Res. 40:D261-70, 2012
runClue  

_Run CLUster Evaluation_

**Description**

Takes in a time-course matrix and test for enrichment of the clustering using cmeans or kmeans clustering algorithm with a reference annotation.

**Usage**

```r
runClue(Tc, annotation, rep = 10, kRange, clustAlg = "cmeans",
        effectiveSize = c(5, 100), pvalueCutoff = 0.05)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tc</td>
<td>a numeric matrix to be clustered. The columns correspond to the time-course</td>
</tr>
<tr>
<td></td>
<td>and the rows correspond to phosphorylation sites.</td>
</tr>
<tr>
<td>annotation</td>
<td>a list with names correspond to kinases and elements correspond to substrates</td>
</tr>
<tr>
<td></td>
<td>belong to each kinase.</td>
</tr>
<tr>
<td>rep</td>
<td>number of times the clustering is to be applied. This is to account for variability</td>
</tr>
<tr>
<td></td>
<td>in the clustering algorithm.</td>
</tr>
<tr>
<td>kRange</td>
<td>the range of k to be tested for clustering.</td>
</tr>
<tr>
<td>clustAlg</td>
<td>the clustering algorithm to be used. The default is cmeans clustering.</td>
</tr>
<tr>
<td>effectiveSize</td>
<td>the size of annotation groups to be considered for calculating enrichment.</td>
</tr>
<tr>
<td></td>
<td>Groups that are too small or too large will be removed from calculating overall enrichment of the clustering.</td>
</tr>
<tr>
<td>pvalueCutoff</td>
<td>a pvalue cutoff for determining which kinase-substrate groups to be included in</td>
</tr>
<tr>
<td></td>
<td>calculating overall enrichment of the clustering.</td>
</tr>
</tbody>
</table>

**Value**

a clue output that contains the input parameters used for evaluation and the evaluation results. Use `ls(x)` to see details of output. 'x' be the output here.

**Examples**

```r
# load the human ES phosphoproteomics data (Rigbolt et al. Sci Signal. 4(164):rs3, 2011)
data(hES)
# load the PhosphoSitePlus annotations (Hornbeck et al. Nucleic Acids Res. 40:D261-70, 2012)data(PhosphoSite)

# make a subset of hES dataset for demonstrating the example in a short time frame
ids <- c("CK2A1", "ERK1", "ERK2", "CDK7",
         "p98RSK", "p70S6K", "PKACA", "CDK1", "DNAPK", "ATM", "CDK2")
hESs <- hES[rownames(hES) %in% unlist(PhosphoSite.human[ids]),]
```
temporalSimu

```
# run CLUE with a repeat of 3 times and a range from 2 to 13
set.seed(2)
clueObj <- runClue(Tc=HESs, annotation=PhosphoSite.human, rep=2, kRange=13)

# visualize the evaluation outcome
Ms <- apply(clueObj$evlMat, 2, mean, na.rm=TRUE)
Ss <- apply(clueObj$evlMat, 2, sd, na.rm=TRUE)
library(Hmisc)
errbar(1:length(Ms), Ms, Ms+Ss, Ms-Ss, cex=1.2, type="b", xaxt="n", xlab="k", ylab="E")
axis(1, at=1:12, labels=paste("k=" , 2:13, sep=""))

# generate the optimal clustering results
best <- clustOptimal(clueObj, rep=10, mfrow=c(3, 4))

# list enriched clusters
best$enrichList

# obtain the optimal clustering object (not run)
# best$clustObj
```

temporalSimu  
Temporal data simulation

Description

This function simulates time-series data using 14 pre-defined temporal profile templates. Type 'temporalSimu' to see the details of the templates.

Usage

```
temporalSimu(seed = unclass(Sys.time()), groupSize, sdd, numGroups)
```

Arguments

- **seed**  
to seed the simulation. Default is current system time.
- **groupSize**  
the number of the temporal profiles to simulate from each template. The total number of profiles will be the number of templates used times the size of each group.
- **sdd**  
the standard deviation to be used to generate randomness for each temporal profile.
- **numGroups**  
number of templates to be used for generating data.

Value

a matrix containing simulated time-series dataset.
Examples

# simulate a time-series data with four distinctive profile groups and each group with
# a size of 500 phosphorylation sites

simulated.temporal <- temporalSimu(seed=1, groupSize=500, sdd=1, numGroups=4)
Index

CLUE (ClueR-package), 2
ClueR-package, 2
clustEnrichment, 2
clustOptimal, 3
enrichmentTest, 5
fuzzPlot, 6
hES, 6
Pathways.KEGG, 7
Pathways.reactome, 7
PhosphoSite.human, 7
PhosphoSite.mouse, 7
runClue, 8
temporalSimu, 9