Package ‘nlnet’

October 13, 2015

Type Package
Title Nonlinear Network Reconstruction and Clustering Based on DCOL
   (Distance Based on Conditional Ordered List)
Version 1.0
Date 2015-10-12
Author Haodong Liu, Tianwei Yu
Maintainer Haodong Liu <liuhaodong0828@gmail.com>
Description It includes three methods: K-profiles clustering, non-linear network reconstruction, and non-linear hierarchical clustering.
License GPL (>= 2)
Imports TSP, igraph, fdrtool, coin, methods, graphics, stats
NeedsCompilation no
Repository CRAN
Date/Publication 2015-10-13 15:51:10

R topics documented:

<table>
<thead>
<tr>
<th>R topic</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>data.gen</td>
<td>1</td>
</tr>
<tr>
<td>KPC</td>
<td>3</td>
</tr>
<tr>
<td>nlhc</td>
<td>4</td>
</tr>
<tr>
<td>nlnet</td>
<td>6</td>
</tr>
</tbody>
</table>

Index 8

---

data.gen Simulated Data Generation

Description
Generating gene matrix as a example of input.
Usage

```r
data.gen(n.genes=100, n.samples=10, n.grps=10, aver.grp.size=10,
  n.fun.types=3, epsilon=1, n.depend=0)
```

Arguments

- **n.genes**: the number of rows of the matrix.
- **n.samples**: the number of columns of the matrix.
- **n.grps**: the number of hidden clusters.
- **aver.grp.size**: average number of genes in a cluster.
- **n.fun.types**: number of function types to use.
- **epsilon**: noise level.
- **n.depend**: data generation dependence structure. can be 0, 1, 2.

Details


Value

- return the data including gene and clustering.

```r
data
grps
```

Author(s)

Tianwei Yu<tyu8@emory.edu>

Examples

```r
## generating a gene matrix with 100 gene and 100 samples per gene.
data.gen(100, 100)
## gene matrix with more than 10 hidden clusters.
data.gen(n.grps=20)
## get the gene matrix from the source of data.
output<-data.gen()
matrix<-output$data
## get the hidden clusters from the source of data.
grps<-output$grp
```
implementation of K-Profiles Clustering

Usage

KPC(dataset, nCluster, maxIter = 100, p.max = 0.2, p.min = 0.05)

Arguments

dataset the data matrix with genes in the row and samples in the column
nCluster the number of clusters K
maxIter the maximum number of iterations
p.max the starting p-value cutoff to exclude noise genes
p.min the final p-value cutoff to exclude noise genes

Value

Return a list about gene cluster and the list of value p

cluster gene cluster
p.list a list of value p

Author(s)

Tianwei Yu <tianwei.yu@emory.edu>

References

http://www.hindawi.com/journals/bmri/aa/918954/

See Also

data.gen

Examples

## generating the data matrix & hiden clusters as a sample
input<-data.gen()

## now input includes data matrix and hiden clusters, so get the matrix as input.
input<-input$data

## set nCluster value to 5
nlhc <- KPC(input, nCluster = 5)

## set maximum number of iteration to 200
nlhc <- KPC(input, nCluster = 10, maxIter = 200)

## get the hidden cluster result from "KPC"
cluster <- nlhc$cluster

## get the list of p
p <- nlhc$p.list

---

### nlhc

**Non-Linear Hierarchical Clustering**

---

**Description**

The non-linear hierarchical clustering based on DCOL

**Usage**

```r
nlhc(array, hamil.method = "nn", concorde.path = NA, use.normal.approx = FALSE, normalization = "standardize", combine.linear = TRUE, use.traditional.hclust = FALSE, method.traditional.hclust = "average")
```

**Arguments**

- `array`: the data matrix with no missing values
- `hamil.method`: the method to find the hamiltonian path.
- `concorde.path`: If using the Concorde TSP Solver, the local directory of the solver
- `use.normal.approx`: whether to use the normal approximation for the null hypothesis.
- `normalization`: the normalization method for the array.
- `combine.linear`: whether linear association should be found by correlation to combine with non-linear association found by DCOL.
- `use.traditional.hclust`: whether traditional agglomerative clustering should be used.
- `method.traditional.hclust`: the method to pass on to hclust() if traditional method is chosen.

**Details**

Hamil.method: It is passed onto the function tsp of library TSP. To use linkern method, the user needs to install concord as instructed in TSP.

use.normal.approx: If TRUE, normal approximation is used for every feature, AND all covariances are assumed to be zero. If FALSE, generates permutation based null distribution - mean vector and a variance-covariance matrix.
normalization: There are three choices - "standardize" means removing the mean of each row and make the standard deviation one; "normal_score" means normal score transformation; "none" means do nothing. In that case we still assume some normalization has been done by the user such that each row has approximately mean 0 and sd 1.

combine.linear: The two pieces of information is combined at the start to initiate the distance matrix.

Value

Returns a hclust object same as the output of hclust(). Reference: help(hclust)

merge

an n-1 by 2 matrix. Row i of merge describes the merging of clusters at step i of the clustering. If an element j in the row is negative, then observation -j was merged at this stage. If j is positive then the merge was with the cluster formed at the (earlier) stage j of the algorithm.

height

a set of n-1 real values, the value of the criterion associated with the clustering method for the particular agglomeration

order

a vector giving the permutation of the original observations suitable for plotting, in the sense that a cluster plot using this ordering and matrix merge will not have crossings of the branches.

labels

labels for each of the objects being clustered

call

the call which produced the result

dist.method

the distance that has been used to create d

height.₀

original calculation of merging height

Author(s)

Tianwei Yu <tianwei.yu@emory.edu>

References


See Also

data.gen

Examples

## generating the data matrix & hiden clusters as a sample
input<-data.gen()
## now input includes data matrix and hiden clusters, so get the matrix as input.
input<-input$data

nlhc.data<-nlhc(input)

##get the merge from the input.
merge<-nlhc.data$merge
nlnet

Non-Linear Network reconstruction from expression matrix

Description

Non-Linear Network reconstruction method

Usage

nlnet(input, min.fdr.cutoff=0.05, max.fdr.cutoff=0.2, conn.proportion=0.007, gene.fdr.plot=FALSE, min.module.size=0, gene.community.method="multilevel", use.normal.approx=FALSE, normalization="standardize", plot.method="communitygraph")

Arguments

input
the data matrix with no missing values.
min.fdr.cutoff
the minimum allowable value of the local false discovery cutoff in establishing links between genes.
max.fdr.cutoff
the maximum allowable value of the local false discovery cutoff in establishing links between genes.
conn.proportion
the target proportion of connections between all pairs of genes, if allowed by the fdr cutoff limits.
gene.fdr.plot
whether plot a figure with estimated densities, distribution functions, and (local) false discovery rates.
min.module.size
the min number of genes together as a module.
gene.community.method
the method for community detection.
use.normal.approx
whether to use the normal approximation for the null hypothesis.
normalization
the normalization method for the array.
plot.method
the method for graph and community plotting.

Details

gene.community.method: It provides three kinds of community detection method: "multilevel", "label.propagation" and "leading.eigenvector".
use.normal.approx: If TRUE, normal approximation is used for every feature, AND all covariances are assumed to be zero. If FALSE, generates permutation based null distribution - mean vector and a variance-covariance matrix.
normalization: There are three choices: "standardize" means removing the mean of each row and make the standard deviation one; "normal_score" means normal score transformation; "none"
means do nothing. In that case we still assume some normalization has been done by the user such that each row has approximately mean 0 and sd 1.

plot.method: It provides three kinds of plotting method: "none" means plotting no graph, "communityGraph" means plotting community with graph, "graph" means plotting graph, "membership" means plotting membership of the community

Value

it returns a graph and the community membership of the graph.

algorithm The algorithm name for community detection

graph An igraph object including edges: Numeric vector defining the edges, the first edge points from the first element to the second, the second edge from the third to the fourth, etc.

community Numeric vector, one value for each vertex, the membership vector of the community structure.

Author(s)

Haodong Liu <liuhaodong0828@gmail.com>

See Also
data.gen

Examples

## generating the data matrix & hidden clusters as a sample
input<-data.gen()
## Now input includes data matrix and hidden clusters, so get the matrix as input.
input<-input$data

## change the plotting method
nlnet(input,plot.method="graph")
## see fdrtool image
nlnet(input,gene.fdr.plot=TRUE)

## use different community detection method
nlnet(input,gene.community.method="label.propagation")

## change the fdr pro to control connections of genes
## adjust the modularity size
nlnet(input,conn.proportion=0.005,min.module.size=10)

## get the result and see it values
result<-nlnet(input)
graph<-result$graph ## a igraph object.
comm<-result$community ## community of the graph
Index

data.gen, 1, 3, 5, 7

KPC, 3

nlhc, 4
nlnet, 6