Escherichia Coli Network

Example for GeneNet 1.2.13 (August 2015) or later


Load GeneNet package

```r
library("GeneNet")
```

## Loading required package: corpcor
## Loading required package: longitudinal
## Loading required package: fdrtool

E. Coli data set (9 time points for 102 genes):

```r
data(ecoli)
dim(ecoli)
```

```r
## [1]  9 102
```

Estimation of partial correlations

Estimate matrix of partial correlation using a shrinkage estimator:

```r
pc = ggm.estimate.pcor(ecoli)
```

```r
## Estimating optimal shrinkage intensity lambda (correlation matrix): 0.1804
```

```r
dim(pc)
```

```r
## [1] 102 102
```

Assign p-values, q-values and empirical posterior probabilities to all 5151 potential edges in the network:

```r
ecoli.edges = network.test.edges(pc, direct=TRUE, fdr=TRUE)
```

```r
## Estimate (local) false discovery rates (partial correlations):
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
## Step 5... prepare for plotting
```
## Estimate (local) false discovery rates (log ratio of spvars):
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
## Step 5... prepare for plotting

dim(ecoli.edges)

## [1] 5151 10

The table lists all edges in the order strength of partial correlations:

ecoli.edges[1:5,

## pcor node1 node2 pval qval prob log.spvar
## 1 0.2318566 51 53 2.220446e-16 3.612205e-13 1.0000000 -0.043537019
## 2 0.2240555 52 53 2.220446e-16 3.612205e-13 1.0000000 -0.040249854
## 3 0.2150782 51 52 2.220446e-16 3.612205e-13 1.0000000 -0.003287165
## 4 0.1732886 7 93 3.108624e-15 3.792816e-12 0.9999945 -0.025293430
## 5 -0.1341889 29 86 1.120811e-09 1.093997e-06 0.9999516 0.022305368

Decide which edges to include in the network

To obtain a graph you need to select top ranking edges according to a suitable criterion. Here are some suggestions:

1. Use local fdr cutoff 0.2, i.e. include all edges with posterior probability of at least 0.8.

ecoli.net = extract.network(ecoli.edges)

dim(ecoli.net)

## [1] 125 11
2. Use local fdr cutoff 0.1, i.e. i.e. include all edges with posterior probability of at least 0.9.

```r
ecoli.net = extract.network(ecoli.edges, cutoff.ggm=0.9, cutoff.dir=0.9)
```

```r
##
## Significant edges: 65
##  Corresponding to 1.26 % of possible edges
##
## Significant directions: 269
##  Corresponding to 5.22 % of possible directions
## Significant directions in the network: 6
##  Corresponding to 9.23 % of possible directions in the network

dim(ecoli.net)
```

```r
## [1] 65 11
```

3. Include a fixed number of edges, say the 70 strongest edges

```r
ecoli.net = extract.network(ecoli.edges, method.ggm="number", cutoff.ggm=70)
```

```r
##
## Significant edges: 70
##  Corresponding to 1.36 % of possible edges
##
## Significant directions: 377
##  Corresponding to 7.32 % of possible directions
## Significant directions in the network: 9
##  Corresponding to 12.86 % of possible directions in the network

dim(ecoli.net)
```

```r
## [1] 70 11
```

Plot network

For plotting we use the graph and Rgraphviz packages from Bioconductor.

```r
library("Rgraphviz")
```

```r
## Loading required package: graph
## Loading required package: grid
```

Create graph object from the list of edges:

```r
node.labels = colnames(ecoli)
gr = network.make.graph(ecoli.net, node.labels, drop.singles=True)
table( edge.info(gr)$dir )
```

```r
##
## forward none
## 9  61
```
sort( node.degree(gr), decreasing=TRUE)

## sucA cspG fixC yheI lacA lacY lacZ asnA eutG yceP yedE ygcE
## 11 8 7 7 6 6 5 5 5 5 5 5
## pspA atpD b1191 b1583 cspA icdA mopB pspB tnaA yaeM ycgX yfaD
## 4 3 3 3 3 3 3 3 3 3 3 3
## dnaG dnaK hupB ibpB yfiA aceB atpG b1963 cchB dnaJ flgD folK
## 2 2 2 2 2 1 1 1 1 1 1 1
## ftsJ gltA lpdA mmpC nuoM sucD yecO ygbD yhdM yjb0
## 1 1 1 1 1 1 1 1 1 1 1 1

Set node and edge attributes for more beautiful graph plotting:

```r
globalAttrs = list()
globalAttrs$edge = list(color = "black", lty = "solid", lwd = 1, arrowsize=1)
globalAttrs$node = list(fillcolor = "lightblue", shape = "ellipse", fixedsize = FALSE)

nodeAttrs = list()
odeAttrs$fillcolor = c('sucA' = "yellow")
edi = edge.info(gr)
edgeAttrs = list()
edgeAttrs$dir = edi$dir # set edge directions
edgeAttrs$lty = ifelse(edi$weight < 0, "dotted", "solid") # negative correlation -> dotted
edgeAttrs$color = ifelse(edi$dir == "none", "black", "red")
edgeAttrs$label = round(edi$weight, 2) # use partial correlation as edge labels

plot(gr, attrs = globalAttrs, nodeAttrs = nodeAttrs, edgeAttrs = edgeAttrs, "fdp")
```