HW 5 Report

Yongchao Qiao

Introduction

Community detection algorithms have been developed for complex subgroup identification. For example, modularity is the extent to which nodes show clustering patterns with greater density within each cluster and less density among different clusters. The value of modularity is from -1 to 1 and the closer to 1, the more the network shows clustering with respect to the given node grouping. The interpretation of modularity is more meaningful when compared to another one.

Background

The problem of community detection requires the partition of a network into communities of densely connected nodes, with the nodes belonging to different communities being only sparsely connected. Precise formulations of this optimization problem are known to be computationally intractable. Several algorithms have therefore been proposed to find reasonably good partitions in a reasonably fast way. This search for fast algorithms has attracted much interest in recent years due to the increasing availability of large network datasets and the impact of networks on everyday life. One can distinguish several types of community detection algorithms: divisive algorithms detect inter-community links and remove them from the network [1]–[3], agglomerative algorithms merge similar nodes/communities recursively [4] and optimization methods are based on the maximization of an objective function [5]–[7]. Modularity has been used to compare the quality of the partitions obtained by different methods, but also as an objective function to optimize.

TASK #1

Prove mathematically the following claim.

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(C_i, C_j).$$

Where m is the number of edges of the network, A_{ij} is the element of the adjacency matrix, C_i and C_j indicate the communities of i and j, k_i and k_j being the degrees of i and j, δ function δ(u,v) is 1 if u = v and 0 otherwise.

Since the only contributions to the sum come from vertex pairs belonging to the same cluster, we can group these contributions together and rewrite the sum over the vertex pairs as a sum over the clusters

$$Q = \sum_{c=1}^{n_c}\left[\frac{l_c}{m} - \left(\frac{d_c}{2m}\right)^2\right].$$

Here, n_c is the number of clusters, l_c the total number of edges joining vertices of module c and d_c the sum of the degrees of the vertices of c.

Proof:

Since δ function δ(u,v) is 1 if u = v and 0 otherwise, the only contribution to the sum come from vertex pairs belonging to the same cluster, then we can just consider the kernels of first and second equations. Here, i and j are representing vertices in the same cluster c.

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta\left(C_i, C_j\right)$$

$= \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)$ since Q = 0 when C_i = Cj, then we consider the Q in same cluster.

$$= \sum_{ij}\frac{A_{ij}}{2m} - \frac{\sum_{ij} k_i k_j}{(2m)^2}$$

Since $l_c = \frac{1}{2}\sum_{ij} A_{ij}$, which is the total number of edges joining vertices of module c. Also, $k_i = \sum_j A_{ij}$ and $k_j = \sum_i A_{ij}$ then $\sum_{ij} k_i k_j = \sum_i k_i \sum_j k_j$. Since k_i and k_j are the degrees of vertex i and j in the same cluster then *both* $\sum_i k_i$ and $\sum_j k_j$ represent the sum of degrees of the vertices of c which means $d_c = \sum_i k_i = \sum_j k_j$.

Thus, for any one cluster c,

$$Q_c = \sum_{ij}\frac{A_{ij}}{2m} - \frac{\sum_{ij} k_i k_j}{(2m)^2} = \frac{2l_c}{2m} - \frac{\sum_i k_i \sum_j k_j}{(2m)^2} = \frac{l_c}{m} - \frac{d_c \times d_c}{(2m)^2} = \frac{l_c}{m} - \left(\frac{d_c}{2m}\right)^2;$$

Therefore, $Q = \sum_{c=1}^{n_c} Q_c = \sum_{c=1}^{n_c}\left[\frac{l_c}{m} - \left(\frac{d_c}{2m}\right)^2\right]$.

TASK #2

Based on the paper by Blondel (2008, see slides 2 and 3), prove mathematically the following equation.

$$\Delta Q = \left[\frac{\sum_{\text{in}} + 2k_{i,\text{in}}}{2m} - \left(\frac{\sum_{\text{tot}} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{\text{in}}}{2m} - \left(\frac{\sum_{\text{tot}}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right],$$

where

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

Proof:

Here, we will focus on the changes of modularity when moving an isolated nodes i into the cluster c.

Initially, we transform the expression of the modularity for vertices in the same cluster. Since vertices are in the same cluster, $Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right) = \sum_{ij}\frac{A_{ij}}{2m} - \frac{\sum_{ij} k_i k_j}{(2m)^2} = \sum_{ij}\frac{A_{ij}}{2m} - \frac{\sum_i k_i \sum_j k_j}{(2m)^2}$. Therefore, in the cluster c, $\sum_{ij} A_{ij}$, which is equal to $\sum_{in}$, is the sum of the weights of the links inside c and both $k_i = \sum_j A_{ij}$ and $k_j = \sum_i A_{ij}$ are equal to $\sum_{tot}$, which is the sum of the weights of the links incident to nodes in c. Then the expression for a single cluster c can be written as $Q = \frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2$.

Then for the original cluster c, the modularity is $Q_c = \frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2$ and for the isolated node i, the modularity is $Q_{node\ i} = -\left(\frac{\sum_{tot}}{2m}\right)^2 = -\left(\frac{k_i}{2m}\right)^2$, since there is no links in the node i, $\sum_{in}$ is equal to 0 and since there is only one node, $\sum_{tot}$ being equal to k_i.

Next, for the new cluster which is combining the original cluster c and node i, the modularity is $Q_{c\,\&\,node\ i} = \frac{\sum_{IN}}{2m} - \left(\frac{\sum_{TOT}}{2m}\right)^2 = \frac{\sum_{in} + 2k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2$, since the sum of the weights of the links inside new cluster C, $\sum_{IN}$, is equal to the value of original $\sum_{in}$ plus the sum of the weights of the links from i to nodes in c, $k_{i,in}$. And since the sum of the weights of the links incident to nodes in new cluster C, $\sum_{TOT}$, is equal to the value of original $\sum_{tot}$ plus the sum of the weights of the links incident to node i, k_i..

Hence, $\Delta Q = Q_{c\,\&\,node\ i} - Q_c - Q_{node\ i}$

$$= \left[\frac{\sum_{IN}}{2m} - \left(\frac{\sum_{TOT}}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2\right] - \left[-\left(\frac{k_i}{2m}\right)^2\right]$$

$$= \left[\frac{\sum_{in} + 2k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2\right] - \left[-\left(\frac{k_i}{2m}\right)^2\right]$$

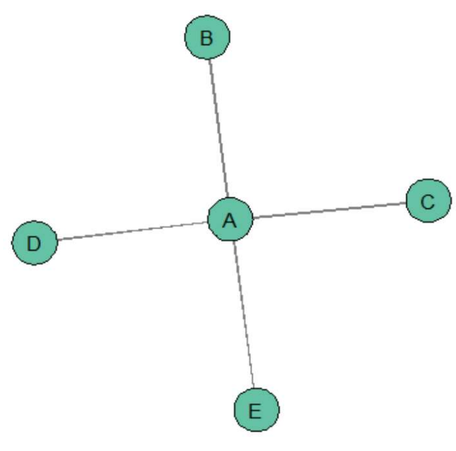
$$= \left[\frac{\sum_{in} + 2k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right].$$

TASK #3

We construct network first:

```
dum1 <- rbind(c(1,2),c(1,2),c(1,3),c(1,4),c(1,5))
star_net <-network(dum1,directed=FALSE)
```

The constructed network is shown below:



Star Graph

For this network, we assign node A in the group one and assign the other four nodes in the group 2.

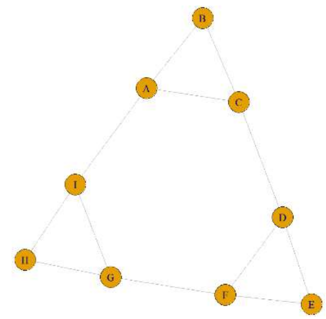
```
modularity(istar,c(1,2,2,2,2))

## [1] -0.5
```

Then in this arrangement, we can calculate the modularity is -0.5 which is not larger than -0.5.

TASK #4

Based on the paper by Blondel (2008, see slides 2 and 3), calculate manually (with stepwise details) the communities detected for the network as below:



Please also complete the related R programming for:

a. Constructing network

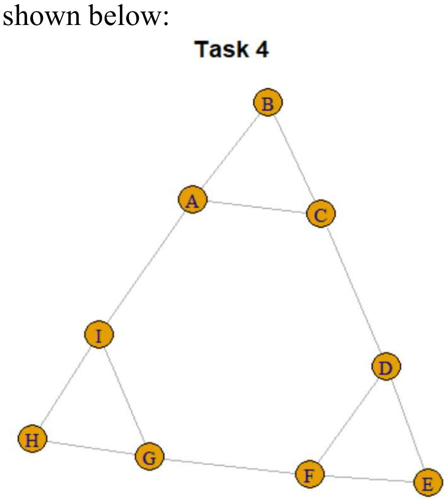
b. Drawing graph

c. Computing modularity

We construct network first:

```
netmat = rbind(c("A","B"),c("A","C"),c("A","I"),c("B","C"),c("C","D"),c("D
","E"),c("D","F"),c("E","F"),c("F","G"),c("G","I"),c("G","H"),c("H","I"))
net = network(netmat, matrix.type = "edgelist",directed = FALSE ) # Use edg
elist to creat the network
network.vertex.names(net) =c("A","B","C","D","E","F","G","H","I")
net
```

The constructed network is shown below:



Then do the calculation:

First, we assign a different community to each node of the network. So, in this initial partition there are as many communities as there are nodes. Then for each node i we consider the neighbors j of i and we evaluate the gain of modularity that would take place by removing i from its community and by placing it in the community of j.

In first step, we can use the formula of ΔQ in Task #2 to calculate the gain of modularity. Here, m is the sum of the weights of all the links in the network and m is equal to 12. And for this formula $\Delta Q_{c,i} = \Delta Q_{i,c}$, so we can only calculate one time for the same new combined cluster.

Thus,

$$\Delta Q_{A,B} = \left[\frac{0+2\times1}{2\times12} - \left(\frac{(0+3)+2}{2\times12}\right)^2\right] - \left[\frac{0}{2\times12} - \left(\frac{(0+3)}{2\times12}\right)^2 - \left(\frac{2}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-5^2+3^2+2^2}{(2\times12)^2} = \frac{36}{24^2} > 0.$$

$$\Delta Q_{A,C} = \left[\frac{0+2\times1}{2\times12} - \left(\frac{(0+3)+3}{2\times12}\right)^2\right] - \left[\frac{0}{2\times12} - \left(\frac{(0+3)}{2\times12}\right)^2 - \left(\frac{3}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-6^2+3^2+3^2}{(2\times12)^2} = \frac{30}{24^2} > 0.$$

$$\Delta Q_{A,I} = \left[\frac{0+2\times1}{2\times12} - \left(\frac{(0+3)+3}{2\times12}\right)^2\right] - \left[\frac{0}{2\times12} - \left(\frac{(0+3)}{2\times12}\right)^2 - \left(\frac{3}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-6^2+3^2+3^2}{(2\times12)^2} = \frac{30}{24^2} > 0.$$

$$\Delta Q_{B,C} = \left[\frac{0+2\times1}{2\times12} - \left(\frac{(0+2)+3}{2\times12}\right)^2\right] - \left[\frac{0}{2\times12} - \left(\frac{(0+2)}{2\times12}\right)^2 - \left(\frac{3}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-5^2+2^2+3^2}{(2\times12)^2} = \frac{36}{24^2} > 0.$$

$$\Delta Q_{C,D} = \left[\frac{0+2\times1}{2\times12} - \left(\frac{(0+3)+3}{2\times12}\right)^2\right] - \left[\frac{0}{2\times12} - \left(\frac{(0+3)}{2\times12}\right)^2 - \left(\frac{3}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-6^2+3^2+3^2}{(2\times12)^2} = \frac{30}{24^2} > 0.$$

$$\Delta Q_{D,E} = \left[\frac{0+2\times1}{2\times12} - \left(\frac{(0+3)+2}{2\times12}\right)^2\right] - \left[\frac{0}{2\times12} - \left(\frac{(0+3)}{2\times12}\right)^2 - \left(\frac{2}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-5^2+3^2+2^2}{(2\times12)^2} = \frac{36}{24^2} > 0.$$

$$\Delta Q_{D,F} = \left[\frac{0+2\times1}{2\times12} - \left(\frac{(0+3)+3}{2\times12}\right)^2\right] - \left[\frac{0}{2\times12} - \left(\frac{(0+3)}{2\times12}\right)^2 - \left(\frac{3}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-6^2+3^2+3^2}{(2\times12)^2} = \frac{30}{24^2} > 0.$$

$$\Delta Q_{E,F} = \left[\frac{0+2\times1}{2\times12} - \left(\frac{(0+2)+3}{2\times12}\right)^2\right] - \left[\frac{0}{2\times12} - \left(\frac{(0+2)}{2\times12}\right)^2 - \left(\frac{3}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-5^2+2^2+3^2}{(2\times12)^2} = \frac{36}{24^2} > 0.$$

$$\Delta Q_{F,G} = \left[\frac{0+2\times1}{2\times12} - \left(\frac{(0+3)+3}{2\times12}\right)^2\right] - \left[\frac{0}{2\times12} - \left(\frac{(0+3)}{2\times12}\right)^2 - \left(\frac{3}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-6^2+3^2+3^2}{(2\times12)^2} = \frac{30}{24^2} > 0.$$

$$\Delta Q_{G,H} = \left[\frac{0+2\times1}{2\times12} - \left(\frac{(0+3)+2}{2\times12}\right)^2\right] - \left[\frac{0}{2\times12} - \left(\frac{(0+3)}{2\times12}\right)^2 - \left(\frac{2}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-5^2+3^2+2^2}{(2\times12)^2} = \frac{36}{24^2} > 0\,.$$

$$\Delta Q_{G,I} = \left[\frac{0+2\times1}{2\times12} - \left(\frac{(0+3)+3}{2\times12}\right)^2\right] - \left[\frac{0}{2\times12} - \left(\frac{(0+3)}{2\times12}\right)^2 - \left(\frac{3}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-6^2+3^2+3^2}{(2\times12)^2} = \frac{30}{24^2} > 0\,.$$

$$\Delta Q_{H,I} = \left[\frac{0+2\times1}{2\times12} - \left(\frac{(0+2)+3}{2\times12}\right)^2\right] - \left[\frac{0}{2\times12} - \left(\frac{(0+2)}{2\times12}\right)^2 - \left(\frac{3}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-5^2+2^2+3^2}{(2\times12)^2} = \frac{36}{24^2} > 0\,.$$

For node A, $\Delta Q_{B,A} = \frac{36}{24^2} > \Delta Q_{C,A} = \frac{30}{24^2} = \Delta Q_{I,A} = \frac{30}{24^2}$, so node A should be assigned into node B's community.

For node C, $\Delta Q_{B,C} = \frac{36}{24^2} > \Delta Q_{A,C} = \frac{30}{24^2} = \Delta Q_{D,C} = \frac{30}{24^2}$,so node C should be assigned into node B's community.

For node B, $\Delta Q_{A,B} = \frac{36}{24^2} = \Delta Q_{C,B} = \frac{36}{24^2}$, so node B should be assigned into node A's and node C's communities.

Hence, node A, B and C should be assigned into one same community.

For node D, $\Delta Q_{E,D} = \frac{36}{24^2} > \Delta Q_{C,D} = \frac{30}{24^2} = \Delta Q_{F,D} = \frac{30}{24^2}$,so node D should be assigned into node E's community.

For node F, $\Delta Q_{E,F} = \frac{36}{24^2} > \Delta Q_{D,F} = \frac{30}{24^2} = \Delta Q_{G,F} = \frac{30}{24^2}$,so node F should be assigned into node E's community.

For node E, $\Delta Q_{D,E} = \frac{36}{24^2} = \Delta Q_{F,E} = \frac{36}{24^2}$, so node E should be assigned into node D's and node F's communities.

Hence, node D, E and F should be assigned into one same community.

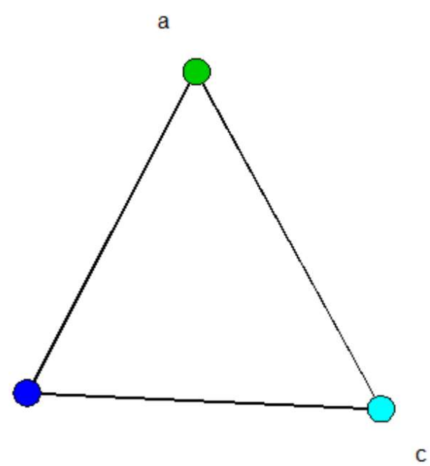
For node G, $\Delta Q_{H,G} = \frac{36}{24^2} > \Delta Q_{I,G} = \frac{30}{24^2} = \Delta Q_{F.G} = \frac{30}{24^2}$, so node G should be assigned into node H's community.

For node I, $\Delta Q_{H,I} = \frac{36}{24^2} > \Delta Q_{G,I} = \frac{30}{24^2} = \Delta Q_{A,I} = \frac{30}{24^2}$,so node I should be assigned into node H's community.

For node H, $\Delta Q_{G,H} = \frac{36}{24^2} = \Delta Q_{I,H} = \frac{36}{24^2}$, so node H should be assigned into node G's and node I's communities.

Hence, node G, H and I should be assigned into one same community.

After step 1, we get a new weighted network with three communities as shown below. Then we will regard each of these communities as one unit and check whether these units can be combined.



Then in second step, we calculate, ΔQ, the gain of modularity. Similarly, we can regard these three communities as three units and follow the formula:

$$\Delta Q = Q_{c1\&c2} - Q_{c1} - Q_{c2}$$

$$= \left[\frac{\sum_{IN}}{2m} - \left(\frac{\sum_{TOT}}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot1}}{2m}\right)^2\right] - \left[\frac{\sum_{in2}}{2m} - \left(\frac{\sum_{tot2}}{2m}\right)^2\right]$$

Here, m is still the sum of the weights of all the links in the network and m is equal to 12. And for this formula $\Delta Q_{ci,cj} = \Delta Q_{cj,ci}$, so we can only calculate one time for the same new combined cluster.

Thus,

$$\Delta Q_{a,b} = \left[\frac{6+6+2\times1}{2\times12} - \left(\frac{(6+6+2)+2}{2\times12}\right)^2\right] - \left[\frac{6}{2\times12} - \left(\frac{(6+2)}{2\times12}\right)^2\right] - \left[\frac{6}{2\times12} - \left(\frac{(6+2)}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-16^2+8^2+8^2}{(2\times12)^2} = \frac{-80}{24^2} < 0\,.$$

$$\Delta Q_{a,c} = \left[\frac{6+6+2\times1}{2\times12} - \left(\frac{(6+6+2)+2}{2\times12}\right)^2\right] - \left[\frac{6}{2\times12} - \left(\frac{(6+2)}{2\times12}\right)^2\right] - \left[\frac{6}{2\times12} - \left(\frac{(6+2)}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-16^2+8^2+8^2}{(2\times12)^2} = \frac{-80}{24^2} < 0\,.$$

$$\Delta Q_{b,c} = \left[\frac{6+6+2\times1}{2\times12} - \left(\frac{(6+6+2)+2}{2\times12}\right)^2\right] - \left[\frac{6}{2\times12} - \left(\frac{(6+2)}{2\times12}\right)^2\right] - \left[\frac{6}{2\times12} - \left(\frac{(6+2)}{2\times12}\right)^2\right]$$

$$= \frac{2}{2\times12} + \frac{-16^2+8^2+8^2}{(2\times12)^2} = \frac{-80}{24^2} < 0\,.$$

Hence, these three new communities can not be combined and the final assignment of this network is that node A, B and C belong to group1, with node D, E and F belonging to group2 and node G, H and I belonging to group3.

```r
modularity(inet,c(rep(1,3),rep(2,3),rep(3,3)))

## [1] 0.4166667
```

Then we can calculate the modularity for this assignment is `0.4166667.`

Appendix

[1] Girvan M and Newman M E J, 2002 Proc. Nat. Acad. Sci. 99 7821
[2] Newman M E J and Girvan M, 2004 Phys. Rev. E 69 026113
[3] Radicchi F, Castellano C, Cecconi F, Loreto V and Parisi D, 2004 Proc. Nat. Acad. Sci. 101 2658
[4] Pons P and Latapy M, 2006 J. Graph Algorithms Appl. 10 191
[5] Clauset A, Newman M E J and Moore C, 2004 Phys. Rev. E 70 066111
[6] Wu F and Huberman B A, 2004 Eur. Phys. J. B 38 331
[7] Newman M E J, 2006 Phys. Rev. E 74 036104

```
library(UserNetR)
library(statnet)
library(RColorBrewer)
```

Task 2

```
dum1 <- rbind(c(1,2),c(1,2),c(1,3),c(1,4),c(1,5)) # Create the edgelist mat
rix

star_net <-network(dum1,directed=FALSE) # Construct the network
network.vertex.names(star_net) =c("A","B","C","D","E") # Add names to the v
ertices

my_pal <- brewer.pal(5,"Set2") # Select the color
gplot(star_net,
      usearrows=FALSE,# Not show arrows
      displaylabels=TRUE, # Show the label
      vertex.cex=2, # Set vertex's size
      vertex.col=my_pal[1],# Set vertex's color
      edge.lwd=0, # Set edge's with
      label.pos = 5, # Set label's position
      edge.col="grey50",# Set edge's color
      xlab="Star Graph" ) # Set title

library(igraph)
library(intergraph)

istar <- asIgraph(star_net) # Transform the network as igraph
modularity(istar,c(1,2,2,2,2)) # Calculate the modularity of assigning node
 A in the group one and assigning the other four nodes in the group 2.
```

Task 3

```
netmat = rbind(c("A","B"),
               c("A","C"),
               c("A","I"),
               c("B","C"),
               c("C","D"),
               c("D","E"),
               c("D","F"),
               c("E","F"),
               c("F","G"),
               c("G","I"),
               c("G","H"),
               c("H","I"))# Createi the edgelist matrix
net = network(netmat, matrix.type = "edgelist",directed = FALSE ) # Use edg
elist to creat the network
network.vertex.names(net) =c("A","B","C","D","E","F","G","H","I")# Add nam
es to the vertices

library(igraph)
library(intergraph)

inet <- asIgraph(net) #Transform the nwtwork as igraph
V(inet)$name <- c("A","B","C","D","E","F","G","H","I")# Add names to the ve
rtices in igraph
set.seed(373)
plot(inet, layout = layout_with_fr, #Set layout as layout_with_fr
     displaylabels = TRUE, # Show the labels
     main = "Task 4")# Set the title

modularity(inet,c(rep(1,3),rep(2,3),rep(3,3)))# Calculate the modularity o
f assigning node A, B and C as group1, with node D, E and F belonging to gro
up2 and node G, H and I belonging to group3.

netmat= rbind(c(0,1,1),
              c(1,0,1),
              c(1,1,0)) # Create the edgelist matrix
rownames(netmat) = c("a","b","c") # Set the rownames
colnames(netmat) = c("a","b","c") # Set the column names
net2 = network(netmat, matrix.type = "adjacency",directed = FALSE) # Creat
the network with adjacency matrix
gplot(net2, vertex.col = c(3,4,5), # Set the colors of vertices
      displaylabels = TRUE, # Show the labels
      usearrows =FALSE, # Not show the arrows
      main = "Three new communities") # set the title
```