Subgroups Chapter 8

Nick Lauerman

Contents

Libraries and data used

Libraries

```
library(UserNetR)
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
library(intergraph)
```

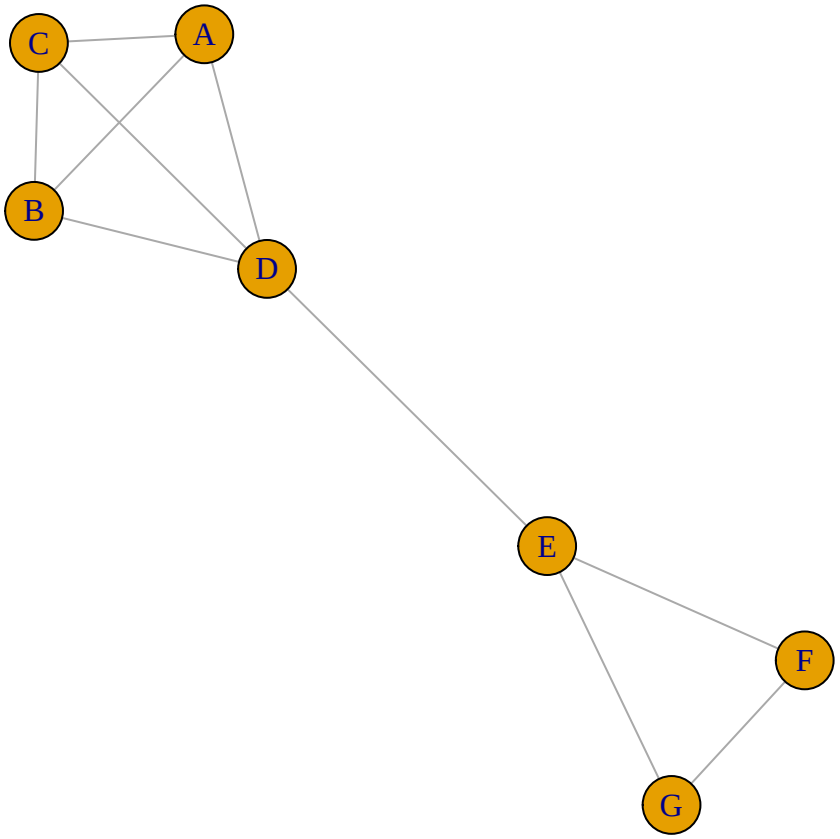
Data

```
data("DHHS")
data("Moreno")
data("Facebook")
```

Social Cohesion

Cliques

```
clqexmp <- graph.formula(A:B:C:D--A:B:C:D,D-E,E-F-G-E)
plot(clqexmp,
     layout = layout_with_fr)
```




```
clique.number(clqexmp)
```

```
## [1] 4
```

```
cliques(clqexmp,
        min = 3)
```

```
## [[1]]
## + 3/7 vertices, named, from 5cbac77:
## [1] B C D
##
## [[2]]
## + 3/7 vertices, named, from 5cbac77:
## [1] E F G
##
## [[3]]
## + 3/7 vertices, named, from 5cbac77:
## [1] A B C
##
## [[4]]
## + 4/7 vertices, named, from 5cbac77:
## [1] A B C D
##
## [[5]]
## + 3/7 vertices, named, from 5cbac77:
## [1] A B D
##
## [[6]]
## + 3/7 vertices, named, from 5cbac77:
## [1] A C D
```

```
maximal.cliques(clqexmp,
                min = 3)
```

```
## [[1]]
## + 3/7 vertices, named, from 5cbac77:
## [1] E F G
##
## [[2]]
## + 4/7 vertices, named, from 5cbac77:
## [1] A B D C
```

```
largest.cliques(clqexmp)
```

```
## [[1]]
## + 4/7 vertices, named, from 5cbac77:
## [1] D A B C
```

```
V(clqexmp)[unlist(largest.cliques(clqexmp))]
```

```
## + 4/7 vertices, named, from 5cbac77:
## [1] D A B C
```

```
g25 <- erdos.renyi.game(25,
                        75,
                        type = "gnm")
g50 <- erdos.renyi.game(50,
                        150,
                        type = "gnm")
g100 <- erdos.renyi.game(100,
```

```
                          300,
                          type = "gnm")
g500 <- erdos.renyi.game(500,
                          1500,
                          type = "gnm")
g1000 <- erdos.renyi.game(1000,
                           3000,
                           type = "gnm")
g5000 <- erdos.renyi.game(5000,
                           15000,
                           type = "gnm")
g10000 <- erdos.renyi.game(10000,
                            30000,
                            type = "gnm")
nodes <- c(25,
           50,
           100,
           500,
           1000,
           5000,
           10000)
lrgclg <- c(clique.number(g25),
            clique.number(g50),
            clique.number(g100),
            clique.number(g500),
            clique.number(g1000),
            clique.number(g5000),
            clique.number(g10000))
numclq <- c(length(cliques(g25,
                           min = 3)),
            length(cliques(g50,
                           min = 3)),
            length(cliques(g100,
                           min = 3)),
            length(cliques(g500,
                           min = 3)),
            length(cliques(g1000,
                           min = 3)),
            length(cliques(g5000,
                           min = 3)),
            length(cliques(g10000,
                           min = 3)))
clqinfo <- data.frame(
  Nodes = nodes,
  Largest = lrgclg,
  number = numclq)
clqinfo
```

```
##   Nodes Largest number
## 1    25       4     45
## 2    50       4     36
## 3   100       3     30
## 4   500       3     39
## 5  1000       3     31
```

```
## 6  5000       3     41
## 7 10000       3     36
```

k-Cores

```
data(DHHS)
iDHHS <- asIgraph(DHHS)
graph.density(iDHHS)
```

```
## [1] 0.312369
```

```
iDHHS <- subgraph.edges(iDHHS,
                        E(iDHHS)[collab > 2])
graph.density(iDHHS)
```

```
## [1] 0.1533688
```

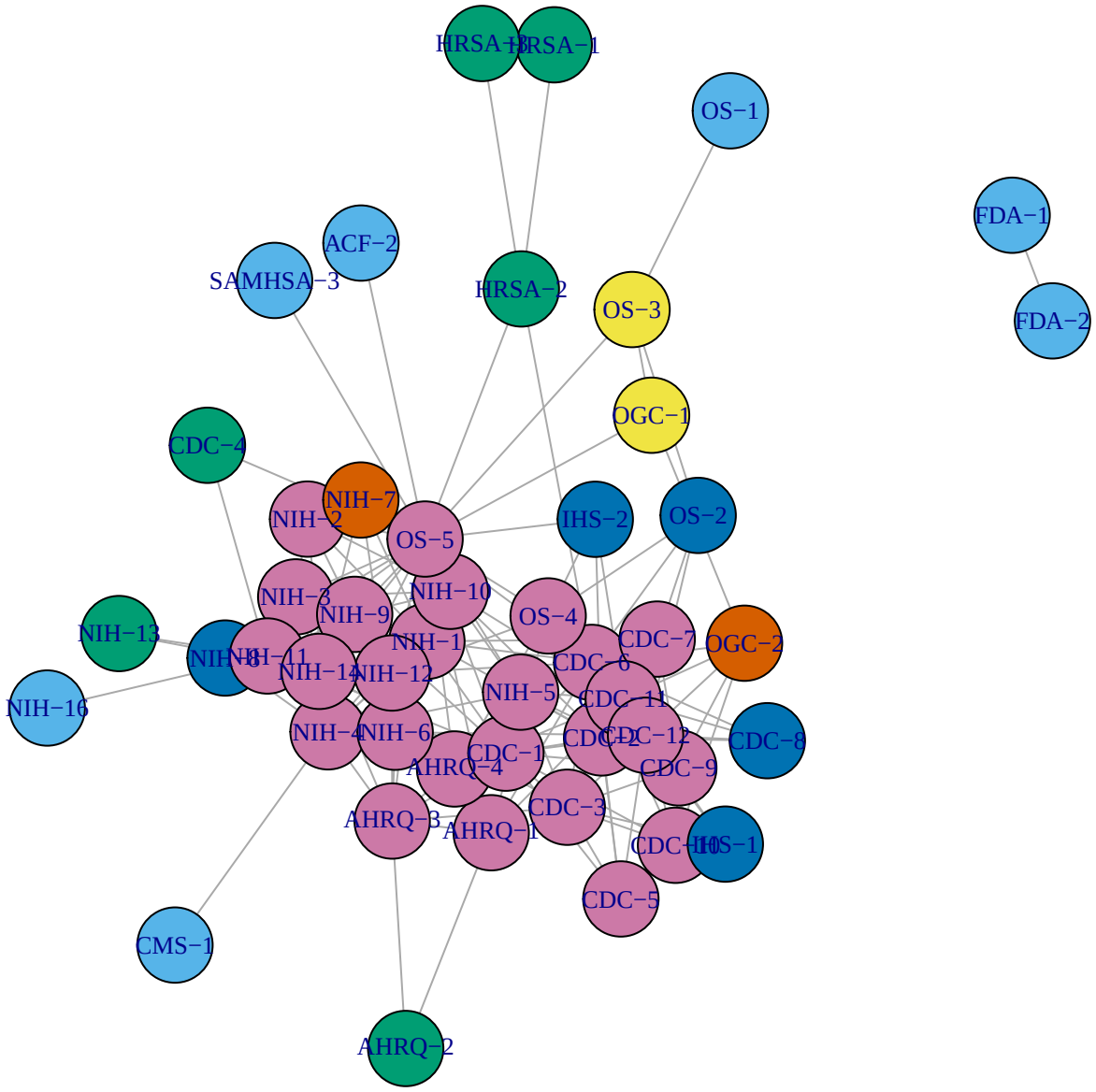
```
coreness <- graph.coreness(iDHHS)
table(coreness)
```

```
## coreness
##  1  2  3  4  5  6
##  7  6  2  5  2 26
```

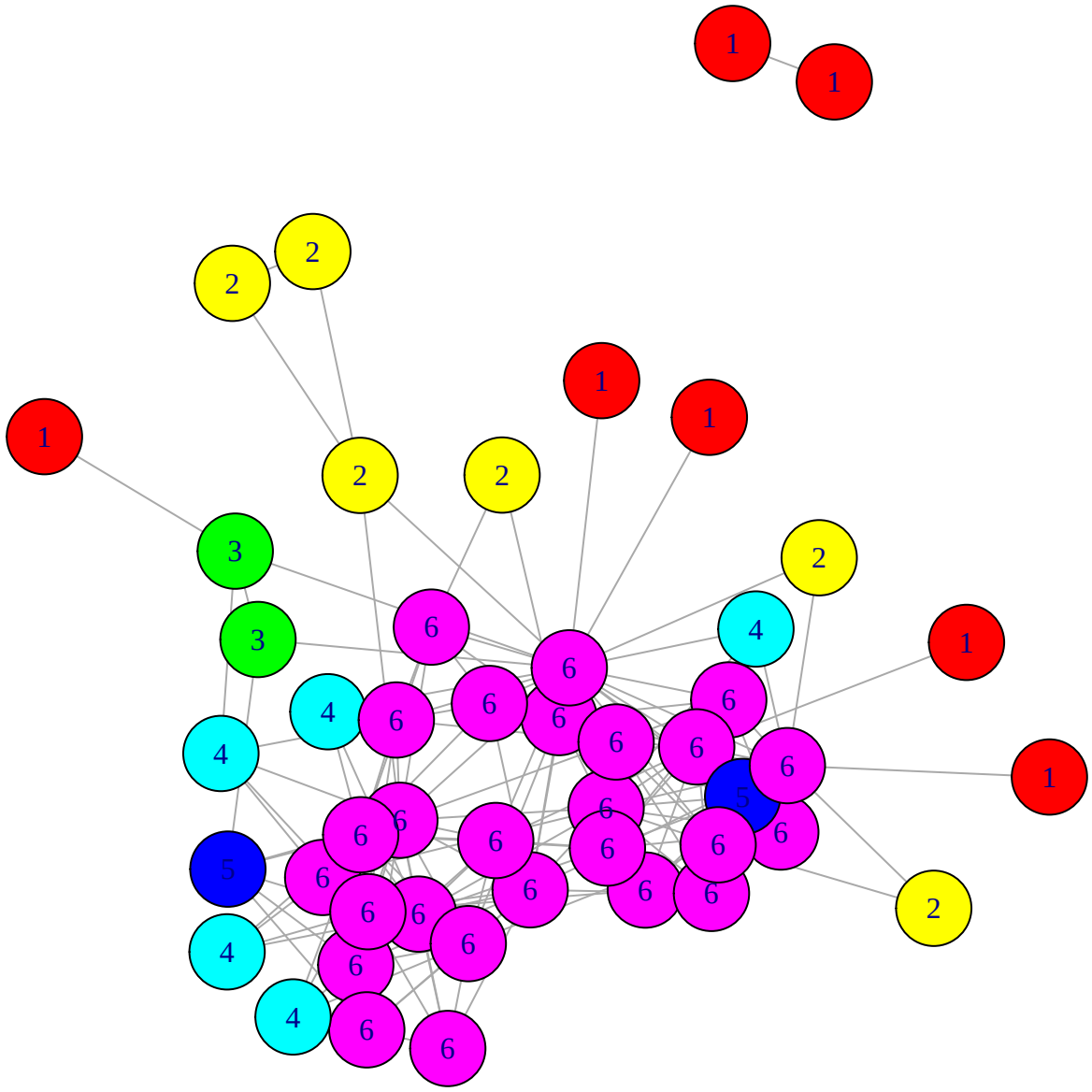
```
maxCoreness <- max(coreness)
maxCoreness
```

```
## [1] 6
```

```
Vname <- get.vertex.attribute(iDHHS,
                              name = "vertex.names",
                              index = V(iDHHS))
V(iDHHS)$name <- Vname
V(iDHHS)$color <- coreness + 1
op <- par(mar = rep(0,4))
plot(iDHHS,
     vertex.label.cex = 0.8)
```



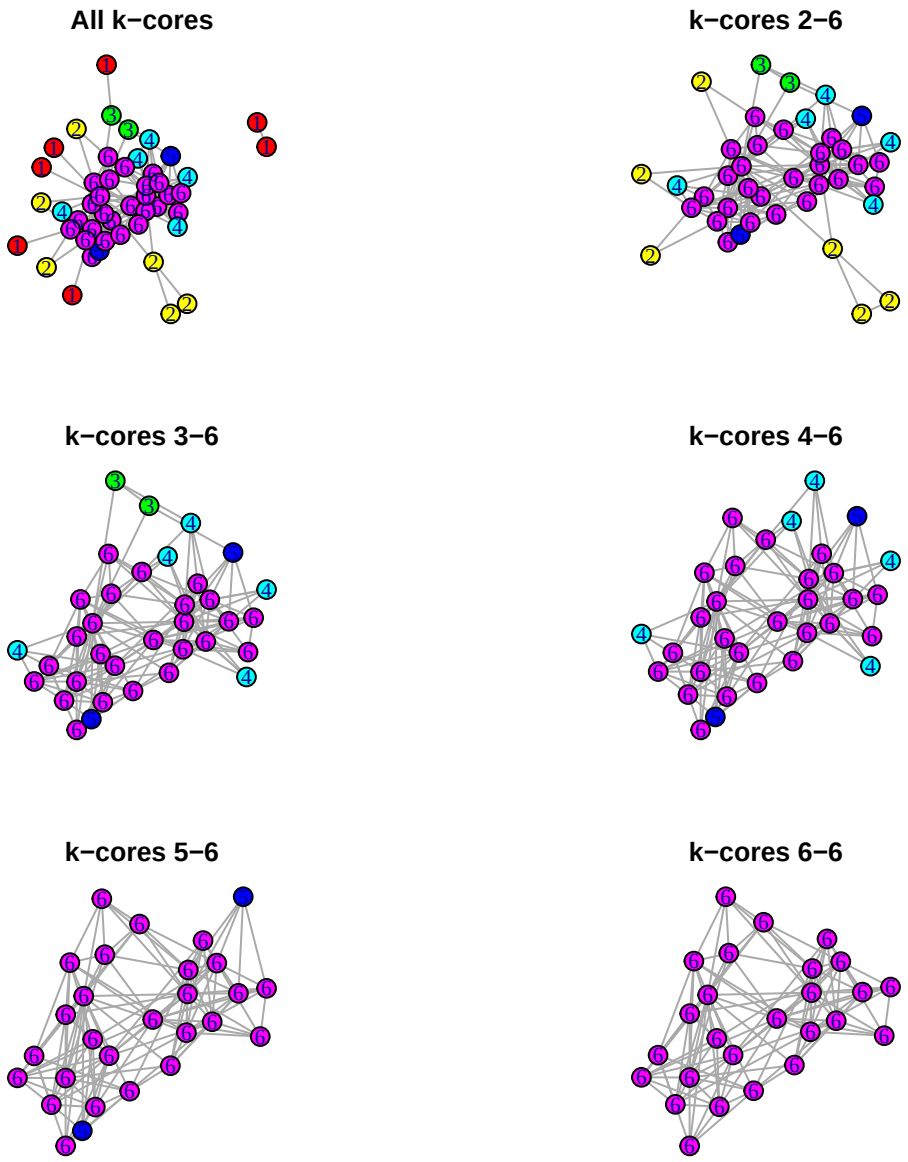

```
par(op)
colors <- rainbow(maxCoreness)
op <- par(mar = rep(0,4))
plot(iDHHS,
     vertex.label = coreness,
     vertex.color = colors[coreness])
```



```
par(op)

V(iDHHS)$name <- coreness
V(iDHHS)$color <- colors[coreness]
iDHHS1_6 <- iDHHS
iDHHS2_6 <- induced.subgraph(iDHHS,
                             vids = which(coreness > 1))
iDHHS3_6 <- induced.subgraph(iDHHS,
                             vids = which(coreness > 2))
iDHHS4_6 <- induced.subgraph(iDHHS,
                             vids = which(coreness > 3))
iDHHS5_6 <- induced.subgraph(iDHHS,
                             vids = which(coreness > 4))
iDHHS6_6 <- induced.subgraph(iDHHS,
```

```
                                vids = which(coreness > 5))
lay <- layout.fruchterman.reingold(iDHHS)
op <- par(mfrow = c(3,2),
          mar = c(3, 0, 2, 0))
plot(iDHHS1_6,
     layout = lay,
     main = "All k-cores")
plot(iDHHS2_6,
     layout = lay[which(coreness > 1), ],
     main = "k-cores 2-6")
plot(iDHHS3_6,
     layout = lay[which(coreness > 2), ],
     main = "k-cores 3-6")
plot(iDHHS4_6,
     layout = lay[which(coreness > 3), ],
     main = "k-cores 4-6")
plot(iDHHS5_6,
     layout = lay[which(coreness > 4), ],
     main = "k-cores 5-6")
plot(iDHHS6_6,
     layout = lay[which(coreness > 5), ],
     main = "k-cores 6-6")
```



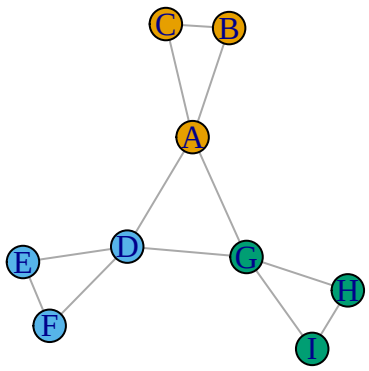

```
par(op)
```

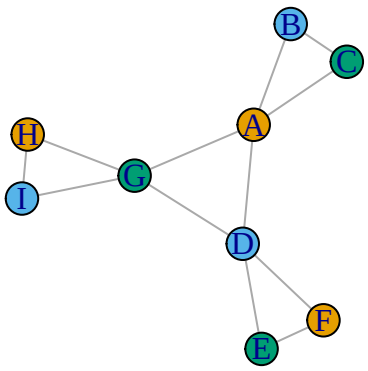
Community Detectiom

Modularity

```
g1 <- graph.formula(A-B-C-A,
                    D-E-F-D,
                    G-H-I-G,
                    A-D-G-A)
V(g1)$grp_good <- c(1,1,1,2,2,2,3,3,3)
V(g1)$grp_bad <-  c(1,2,3,2,3,1,3,1,2)
op <- par(mfrow = c(1,2))
```

```
plot(g1,
     vertex.color = (V(g1)$grp_good),
     vertex.size = 20,
     main = "Good Grouping")
plot(g1,
     vertex.color = (V(g1)$grp_bad),
     vertex.size = 20,
     main = "Bad Grouping")
```






```
par(op)
modularity(g1,
           V(g1)$grp_good)
```

```
## [1] 0.4166667
```

```
modularity(g1,
           V(g1)$grp_bad)
```

```
## [1] -0.3333333
```

```
data(DHHS)
iDHHS <- asIgraph(DHHS)
table(V(iDHHS)$agency)
```

```
##
##  0  1  2  3  4  5  6  7  8  9 10
##  2  4 12  2  2  3  2 16  3  5  3
```

```
V(iDHHS)[1:10]$agency
```

```
##  [1] 0 0 1 1 1 1 2 2 2 2
```

```
modularity(iDHHS,
           (V(iDHHS)$agency + 1))
```

```
## [1] 0.1402264
```

```
data("Moreno")
iMoreno <- asIgraph(Moreno)
table(V(iMoreno)$gender)
```

```
##
##  1  2
## 16 17
```

```
modularity(iMoreno,
           V(iMoreno)$gender)
```

```
## [1] 0.4761342
```

```
data("Facebook")
levels(factor(V(Facebook)$group))
```

```
## [1] "B" "C" "F" "G" "H" "M" "S" "W"
```

```
grp_mun <- as.numeric(factor(V(Facebook)$group))
modularity(Facebook, grp_mun)
```

```
## [1] 0.6145798
```

Community Detection Algorithms

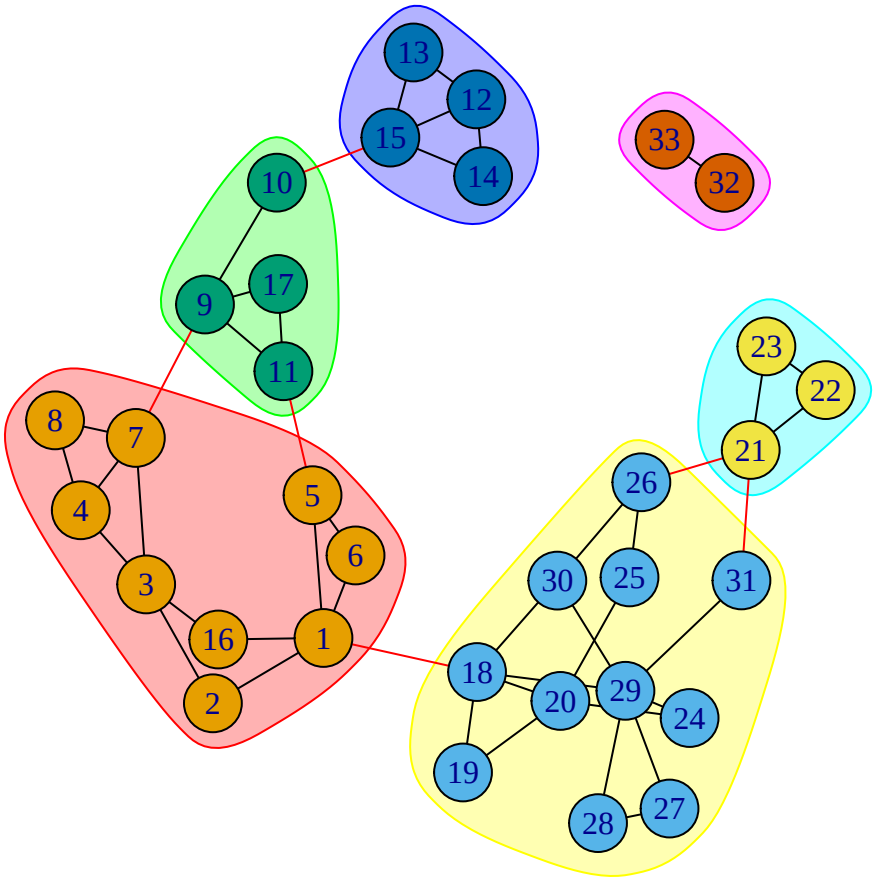
```
cw <- cluster_walktrap(iMoreno)
membership(cw)
```

```
##  [1] 1 1 1 1 1 1 1 1 3 3 3 5 5 5 5 1 3 2 2 2 4 4 4 2 2 2 2 2 2 2 2 2 6 6
```

```
modularity(cw)
```

```
## [1] 0.6181475
```

```
plot(cw, iMoreno)
```




```
cw <- cluster_walktrap(iDHHS)
modularity(cw)
```

```
## [1] 0.1653754
```

```
membership(cw)
```

```
##  [1] 4 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3
## [39] 3 3 3 3 3 2 2 2 2 1 2 1 1 1 1 1
```

```
table(V(iDHHS)$agency,
      membership(cw))
```

```
##
##       1  2  3  4  5
##   0   0  0  0  1  1
```

```
##   1   4  0  0  0  0
##   2  12  0  0  0  0
##   3   2  0  0  0  0
##   4   2  0  0  0  0
##   5   3  0  0  0  0
##   6   2  0  0  0  0
##   7   0  0 16  0  0
##   8   0  3  0  0  0
##   9   3  2  0  0  0
##   10  3  0  0  0  0
```

```
data("Bali")
iBali <- asIgraph(Bali)

cw <- cluster_walktrap(iBali)
modularity(cw)
```

```
## [1] 0.2830688
```

```
membership(cw)
```

```
##  [1] 2 1 2 1 2 2 1 2 2 3 3 3 3 3 2 2 2
```

```
ceb <- cluster_edge_betweenness(iBali)
modularity(ceb)
```

```
## [1] 0.2387251
```

```
membership(ceb)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 2 2 2 2 2 1 1 1
```

```
cs <- cluster_spinglass(iBali)
modularity(cs)
```

```
## [1] 0.2966742
```

```
membership(cs)
```

```
##  [1] 3 2 3 1 3 3 2 3 3 1 1 1 1 1 3 3 3
```

```
cfg <- cluster_fast_greedy(iBali)
modularity(cfg)
```

```
## [1] 0.2629126
```

```
membership(cfg)
```

```
##  [1] 2 2 1 2 1 2 2 1 1 3 3 3 3 3 1 1 1
```

```
clp <- cluster_label_prop(iBali)
modularity(clp)
```

```
## [1] 0.2387251
```

```
membership(clp)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 2 2 2 2 2 1 1 1
```

```
cle <- cluster_leading_eigen(iBali)
modularity(cle)
```

```
## [1] 0.2750063
```

```
membership(cle)
```

```
##  [1] 1 1 1 2 1 1 2 1 1 2 2 2 2 2 1 1 1
```

```
cl <- cluster_louvain(iBali)
modularity(cl)
```

```
## [1] 0.2966742
```

```
membership(cl)
```

```
##  [1] 3 1 3 2 3 3 1 3 3 2 2 2 2 2 3 3 3
```

```
co <- cluster_optimal(iBali)
modularity(co)
```

```
## [1] 0.2966742
```

```
membership(co)
```

```
##  [1] 1 2 1 3 1 1 2 1 1 3 3 3 3 3 1 1 1
```

```
table(V(iBali)$role,
      membership(cw))
```

```
##
##      1 2 3
##   BM 0 5 0
##   CT 1 2 0
##   OA 2 1 0
##   SB 0 1 1
##   TL 0 0 4
```

```
compare(as.numeric(factor(V(iBali)$role)),
                           cw,
                           method = "adjusted.rand")
```

```
## [1] 0.3504908
```

```
compare(cw,
        ceb,
        method = "adjusted.rand")
```

```
## [1] 0.6155779
```

```
compare(cw,
        cs,
        method = "adjusted.rand")
```

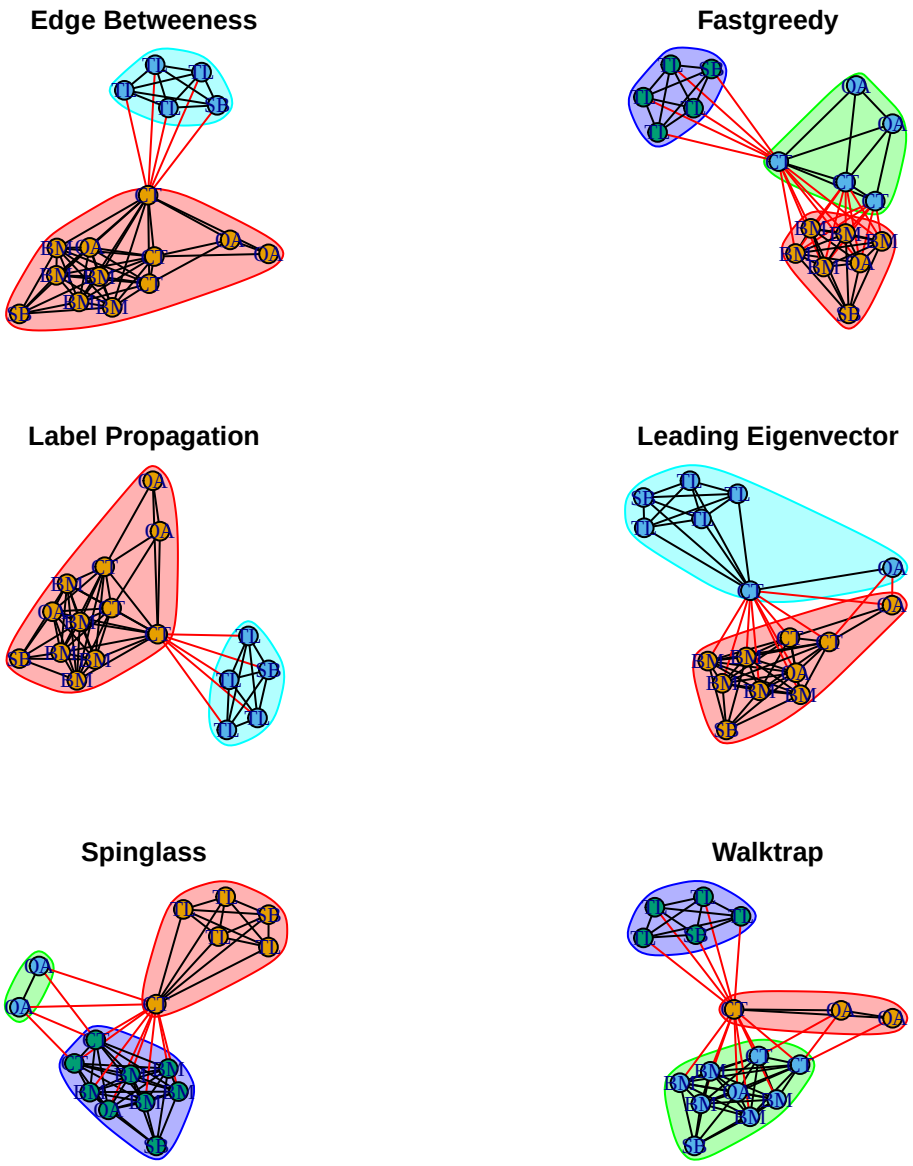
```
## [1] 0.8898148
```

```
compare(cw,
        cfg,
        method = "adjusted.rand")
```

```
## [1] 0.6691802
```

```
op <- par(mfrow = c(3,2),
          mar = c(3, 0, 2, 0))
plot(ceb,
```

```
     iBali,
     vertex.label = V(iBali)$role,
     main = "Edge Betweeness")
plot(cfg,
     iBali,
     vertex.label = V(iBali)$role,
     main = "Fastgreedy")
plot(clp,
     iBali,
     vertex.label = V(iBali)$role,
     main = "Label Propagation")
plot(cle,
     iBali,
     vertex.label = V(iBali)$role,
     main = "Leading Eigenvector")
plot(cs,
     iBali,
     vertex.label = V(iBali)$role,
     main = "Spinglass")
plot(cw,
     iBali,
     vertex.label = V(iBali)$role,
     main = "Walktrap")
```


Edge Betweeness
Fastgreedy
Label Propagation
Leading Eigenvector
Spinglass
Walktrap

```
par(op)
```