

Social Network Analysis for Project SMEs using with RStudio. (work in progres)

Wieteska, Michal

- The of this project is an analysis of SMEs (Subject Matter Experts) invited to the project to provide specialty expertise with regards to ptoejct scope/ deliverabled. The approach take here isusing “social network analysis” concept (see wikipedia link below) with the aim to understand a community by mapping the relationships. These relationships connect them as a network, and then trying to draw out key individuals, groups within the network, associations between the individuals, outline connections intensivity etc. A network is simply a number of nodes that are connected by links. These nodes are people (here SMEs) and the links are any social connection between them; here based on interaction justified by working on the same project/ theme.
- A general concept: see wikipedia: https://en.wikipedia.org/wiki/Social_network_analysis
- for key therms explanation see wikipedia [https://en.wikipedia.org/wiki/Vertex_\(graph_theory\)](https://en.wikipedia.org/wiki/Vertex_(graph_theory))

Starting with RStudio

Proper pdf encoding

Remove all lists

```
ls()  
  
## character(0)  
  
rm(list = ls())
```

Install

```
install.packages("igraph")
```

Load needed packages

```
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 3.6.3
```

```
##
```

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

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

```
##
```

```
##      decompose, spectrum
```

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

```
##
```

```
##      union
```

Read data - manual file location/ selection

```
SMEnetw <- read.csv(file.choose(), header = T, sep = ";", fileEncoding="UTF-8-BOM")
SME_analysis <- data.frame(SMEnetw$SMEstart, SMEnetw$SMEend, SMEnetw$ContactFrequency)
library(igraph)
SMEnetwork <- graph.data.frame(SME_analysis, directed = TRUE)
```

Network analysis key definitions & functions (igraph package)

- Data structure: `str(SMEnetw)`
- Edge (link between nodes), example: `edge_density(SMEnetwork, loops = T)`, example: `ecount(SMEnetwork)`
- Vertex (node apex, point of connection for links), example: `vcount(SMEnetwork)`
- Reciprocity, example: `reciprocity(SMEnetwork)`
- Closeness, example: `closeness(SMEnetwork, mode='all', weights = NA)`
- Betweenness - a Unique links to others in the network i.e. who gets most connections, example: `betweenness(SMEnetwork, directed=TRUE, weights = NA)`
- Degree - Connected to many individuals, number of connections: `degree(SMEnetwork)`
- Matrix - contingency table: `SMEnetwork[]`
- See number of
 - all connections: `degree(SMEnetwork, mode='all')`
 - incoming connections: `degree(SMEnetwork, mode='in')`
 - outgoing connections: `degree(SMEnetwork, mode='out')`
 - see a network diameter: `diameter(SMEnetwork, directed=F, weights=NA)`

Imported data view to understand data structure

```
str(SMEnetw)
```

```
## 'data.frame':   264 obs. of  3 variables:
##  $ SMEstart      : Factor w/ 16 levels "A","B","C","D",...: 1 13 4 6 5 12 7 15 2 10 ...
##  $ SMEend        : Factor w/ 18 levels "A","B","D","E",...: 18 2 11 17 7 18 2 11 17 7 ...
##  $ ContactFrequency: int   4 2 5 7 8 5 3 5 7 8 ...
```

```
head(SMEnetwork)
```

```
## 6 x 20 sparse Matrix of class "dgCMatrix"

##      [[ suppressing 20 column names 'A', 'S', 'D' ... ]]

##
## A . 1 2 1 2 2 3 1 1 1 . 1 . . . . .
## S 1 . 1 1 2 4 3 4 1 1 1 1 1 . . . 5 . . .
## D . 2 . . 2 6 3 2 2 1 3 1 . . . . 1 . 2 .
## F 2 4 1 . . 2 3 1 1 2 3 1 . . . 1 3 . . .
## E 2 2 2 2 . 1 2 2 . 1 5 1 . . . 1 1 3 . .
## R 3 3 2 2 . . 2 3 2 . 2 2 . . . . . 2 2
```

```
tail(SMEnetwork)
```

```
## 6 x 20 sparse Matrix of class "dgCMatrix"

##      [[ suppressing 20 column names 'A', 'S', 'D' ... ]]

##
## E 2 2 2 2 . 1 2 2 . 1 5 1 . . . 1 1 3 . .
## R 3 3 2 2 . . 2 3 2 . 2 2 . . . . . 2 2
## G 1 4 3 2 2 1 . 1 3 2 . 2 . . . 1 1 1 . .
## V . 1 2 3 1 1 . . 1 2 3 . 1 . . . . .
## B . . 2 2 1 1 1 . . 2 3 1 . . . . .
## N 1 3 . 2 . 2 1 . . . 1 2 1 . . . . .
```

```
degree(SMEnetwork)
```

```
##  A  S  D  F  E  R  G  V  B  N  U  H  J  C  W  O  M  Q  T  I
## 27 50 46 44 42 54 49 34 25 28 52 29 18  1  1  4 12  5  5  2
```

```
SMEnetwork[]
```

```
## 20 x 20 sparse Matrix of class "dgCMatrix"

##      [[ suppressing 20 column names 'A', 'S', 'D' ... ]]

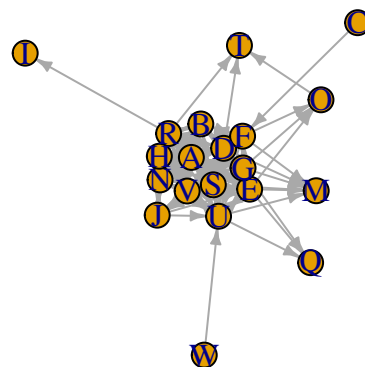
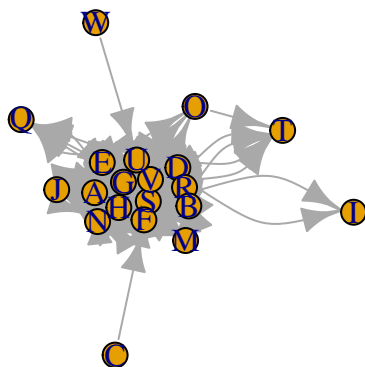
##
## A . 1 2 1 2 2 3 1 1 1 . 1 . . . . .
## S 1 . 1 1 2 4 3 4 1 1 1 1 1 . . . 5 . . .
## D . 2 . . 2 6 3 2 2 1 3 1 . . . . 1 . 2 .
## F 2 4 1 . . 2 3 1 1 2 3 1 . . . 1 3 . . .
## E 2 2 2 2 . 1 2 2 . 1 5 1 . . . 1 1 3 . .
## R 3 3 2 2 . . 2 3 2 . 2 2 . . . . . 2 2
## G 1 4 3 2 2 1 . 1 3 2 . 2 . . . 1 1 1 . .
## V . 1 2 3 1 1 . . 1 2 3 . 1 . . . . .
## B . . 2 2 1 1 1 . . 2 3 1 . . . . .
```

```
## N 1 3 . 2 . 2 1 . . . 1 2 1 . . . . .
## U 1 3 5 . 4 4 3 3 . . . 3 . . . 1 1 . .
## H 1 . 1 3 . 3 1 2 1 1 . . 1 . . . . .
## J . 1 . 1 3 2 3 . . 2 2 . . . . .
## C . . . 1 . . . . . . . . . . .
## W . . . . . . . . 1 . . . . .
## O . . . . . . . . . . . . 1 .
## M . . . . . . . . . . . . . . .
## Q . . . . . . . . . . . . . . .
## T . . . . . . . . . . . . . . .
## I . . . . . . . . . . . . . . .
```

A basic vs simplified graph (removed loops in the graph, multiple edges)

```
par(mfrow=c(1,2))
plot(SMENetwork)

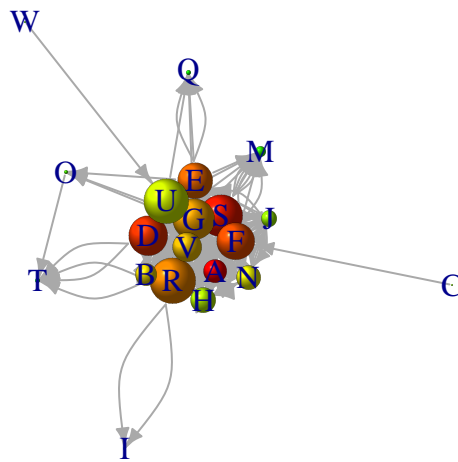
SMENetwork_n1 <- simplify(SMENetwork, remove.multiple = T, remove.loops = T)
plot(SMENetwork_n1, edge.arrow.size=.4) # no loops
```



Modified basic plot

```
V(SMENetwork)$ContactFrequency <- degree(SMENetwork)

plot(SMENetwork,
     vertex.color = rainbow(53),
     vertex.shape="sphere",
     vertex.size = V(SMENetwork)$ContactFrequency*0.4,
     edge.arrow.size = 0.5,
     layout=layout.fruchterman.reingold)
```

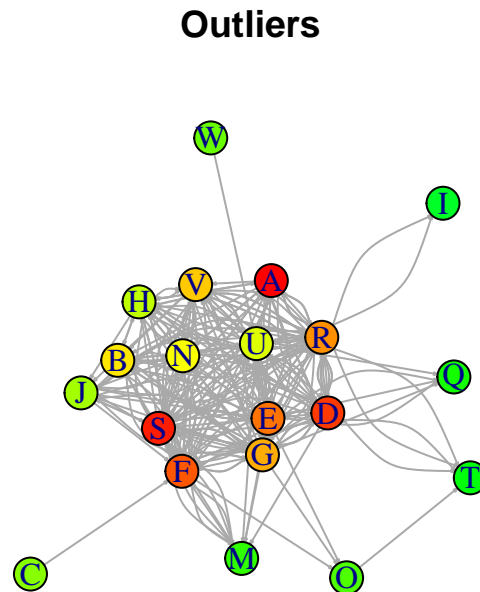


- Network very dense with a couple of outliers, seems like a key interactions are happening within 1 group

A view on outliers (probe)

```
plot(SMENetwork,
     vertex.color = rainbow(53),
     # vertex.size = V(SMENetwork)$degree*0.4,
     edge.arrow.size = 0.1,
     layout=layout.kamada.kawai,
```

```
main="Outliers",
cex.main=0.2,
font.main=2)
```



* Comments: 7 outlier identified with diversified number of connections starting from 1

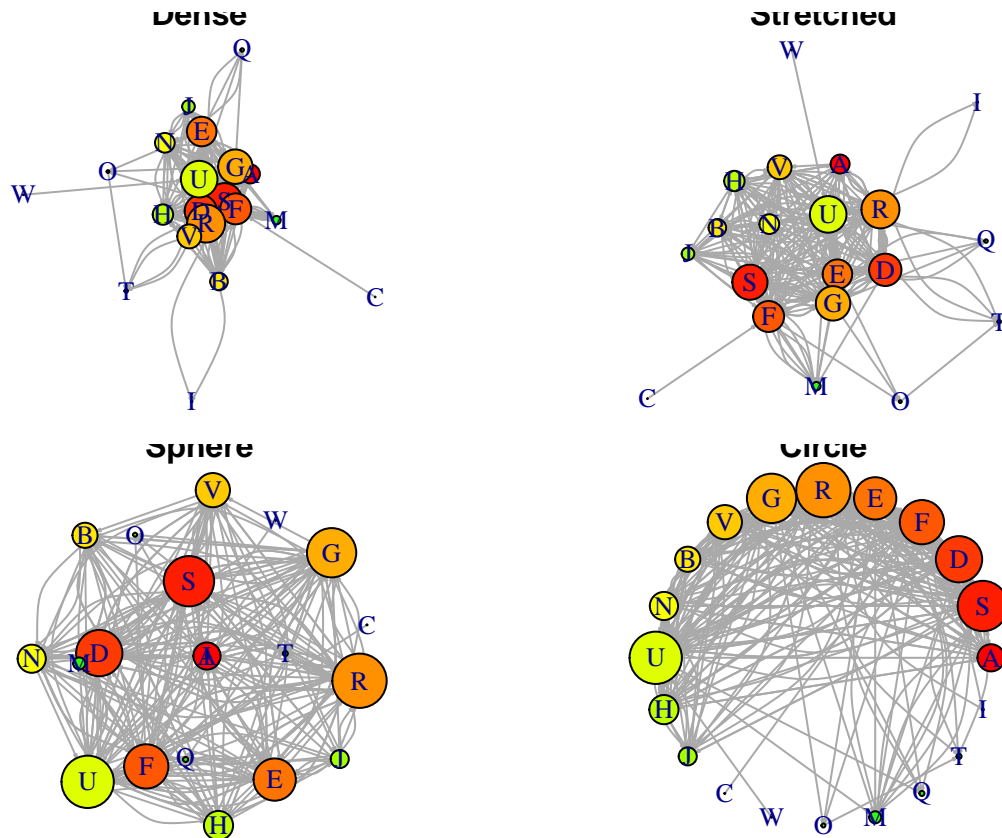
4 different view to get more insight

```
par(mfrow=c(2,2))
par(mar=c(0.25,0.25,0.1,0.1))
plot(SMEnetwork,
     vertex.color = rainbow(53),
     vertex.size = V(SMEnetwork)$ContactFrequency*0.4,
     edge.arrow.size = 0.1,
     main="Dense",
     cex.main=0.5,
     font.main=1,
     layout=layout.graphopt)
plot(SMEnetwork,
     vertex.color = rainbow(53),
     vertex.size = V(SMEnetwork)$ContactFrequency*0.4,
     edge.arrow.size = 0.1,
     main="Stretched",
     cex.main=0.5,
```

```

font.main=1,
layout=layout.kamada.kawai)
plot(SMEnetwork,
vertex.color = rainbow(53),
vertex.size = V(SMEnetwork)$ContactFrequency*0.6,
edge.arrow.size = 0.1,
main="Sphere",
cex.main=0.5,
font.main=1,
layout=layout.sphere)
plot(SMEnetwork,
vertex.color = rainbow(53),
vertex.size = V(SMEnetwork)$ContactFrequency*0.6,
edge.arrow.size = 0.1,
main="Circle",
cex.main=0.5,
font.main=1,
layout=layout.circle)

```

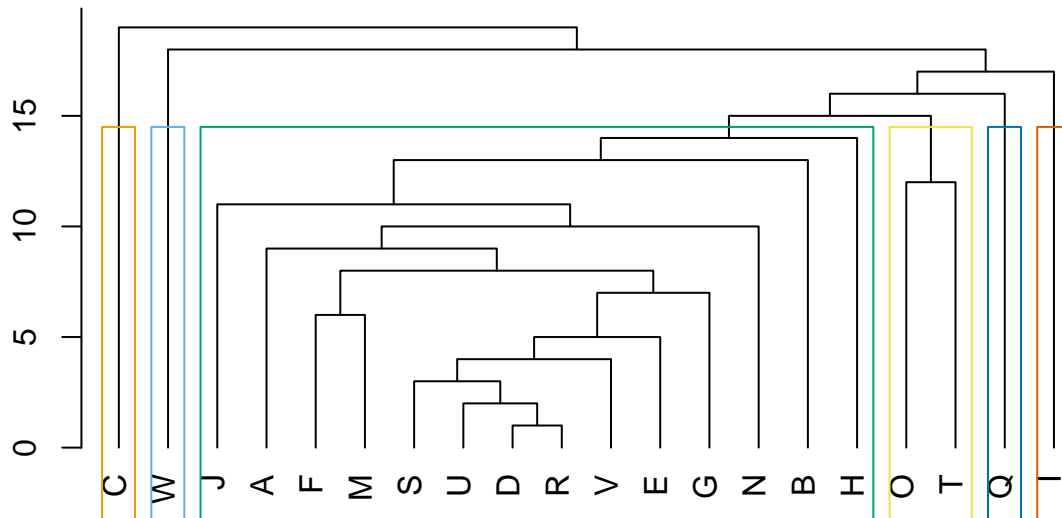


* Comments: visually it looks like D, G, R, S and U are actors with the most connections (interactions)

Community detection (identify, outline graphically community areas)

by dendrogram

```
par(bg="white") # set background  
ceb <- cluster_edge_betweenness(SMENetwork)  
dendPlot(ceb, mode="hclust") # plot(hcd, type = "triangle", ylab = "Height")
```



* Comments: a big group of SMEs sit close interaction identified (green outline)

by network

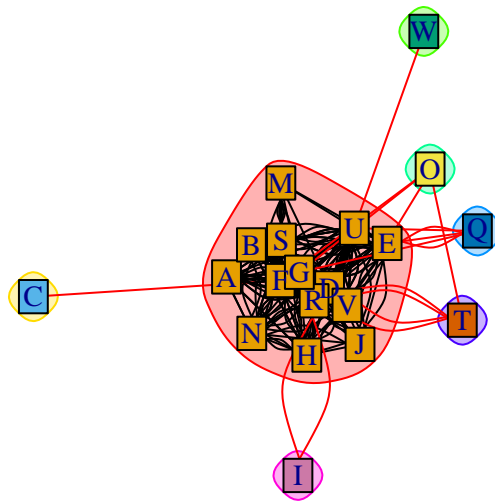
```
SMENetwork1 <- igraph::as_data_frame(SMENetwork)  
SMENetwork2 <- graph.data.frame(SMENetwork1, directed = F)  
cnet <- cluster_edge_betweenness(SMENetwork2)  
  
plot(cnet,  
     SMENetwork,  
     vertex.size = 13,  
     vertex.label.cex = 0.9,
```



```

edge.arrow.size=0.1,
vertex.shape="rectangle",
vertex.label.dist=0.0,
edge.arrow.mode=0,
vertex.size = V(SMEnetwork2)$ContactFrequency*0.2,
layout=layout.graphopt)

```



Outlining paths - 3 views

```

par(mfrow=c(1,2))

clp <- cluster_label_prop(SMEnetwork)

plot(clp, SMEnetwork,
     edge.arrow.size = 0.1,
     main="View A",
     cex.main=0.5,
     font.main=1,)

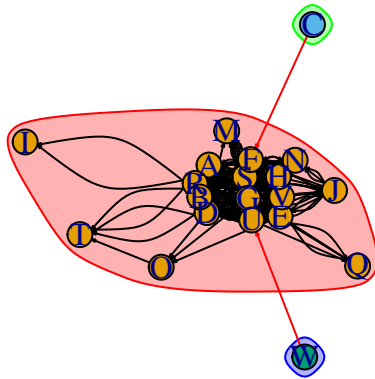
SMEnetwork1 <- igraph::as_data_frame(SMEnetwork)

```

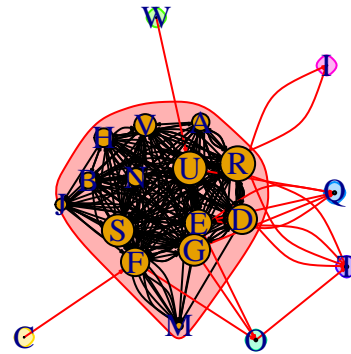
```
SMENetwork2 <- graph.data.frame(SMENetwork1, directed = F)
cnet <- cluster_edge_betweenness(SMENetwork2)
```

```
plot(cnet,
     SMENetwork,
     vertex.color = rainbow(53),
     vertex.size = V(SMENetwork)$ContactFrequency*0.4,
     edge.arrow.size = 0.1,
     main="View B",
     cex.main=0.1,
     font.main=1,
     layout=layout.kamada.kawai)
```

View A



View B



... cont.

```
# greedy method
coords = layout_with_fr(SMENetwork)

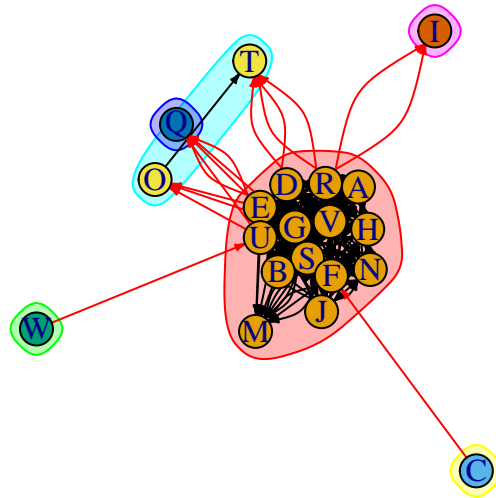
c3 = cluster_edge_betweenness(SMENetwork * 1.5)

# modularity measure
modularity(c3)
```

```
## [1] 0.006011823
```

```
# plot communities with shaded regions
plot_shaded_view <- plot(c3, SMEnetwork, edge.arrow.size = 0.25, layout=coords, main="View C"); plot_sh
```

View C



```
## NULL
```

Identification of Hubs and Authorities

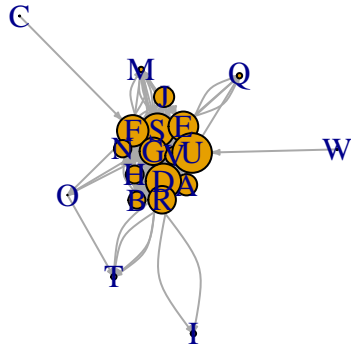
```
hs <- hub_score(SMEnetwork, weights=NA)$vector
as <- authority_score(SMEnetwork, weights=NA)$vector

par(mfrow=c(1,2))

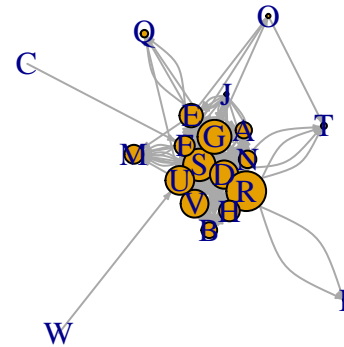
plot(SMEnetwork, vertex.size=hs*25, main="Hubs", edge.arrow.size = 0.2)

plot(SMEnetwork, vertex.size=as*25, main="Authorities", edge.arrow.size = 0.2)
```

Hubs



Authorities



```
#distances(SMENetwork)
```

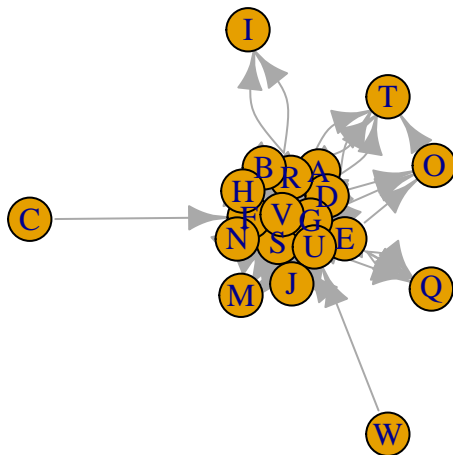
Beter zoom

```
par(mfrow=c(1,2), mar=c(0,0,0,0))

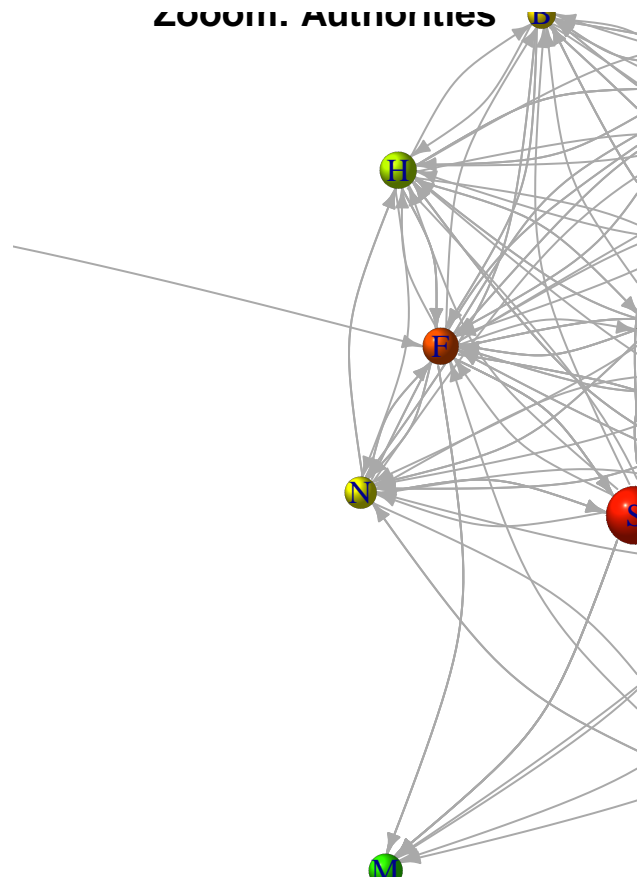
l <- layout_with_fr(SMENetwork)
l <- norm_coords(l, ymin=-1, ymax=1, xmin=-1, xmax=1)

plot(SMENetwork, main="Highlevel: Authorities", rescale=F, layout=l*0.7, cex.main=0.1)
plot(SMENetwork, vertex.shape="sphere", vertex.color = rainbow(53), vertex.size=as*25, main="Zoom: Autl")
```

Highlevel. Authorities



Zoom. Authorities



```
# edge.color=c("gold", "blue", "tomato", "grey", "red", "yellowgreen", "black"), rescale=F, layout=l*
```

Additional analysis

```
degree(SMEnetwork)
```

```
## A S D F E R G V B N U H J C W O M Q T I
## 27 50 46 44 42 54 49 34 25 28 52 29 18 1 1 4 12 5 5 2
```

```
reciprocity(SMEnetwork)
```

```
## [1] 0.530303
```

```
closeness(SMEnetwork, mode='all', weights = NA)
```

```
## A S D F E R G
## 0.03703704 0.04000000 0.04000000 0.04347826 0.04347826 0.04166667 0.04347826
## V B N U H J C
## 0.03846154 0.03703704 0.03846154 0.04347826 0.03846154 0.03448276 0.02439024
## W O M Q T I
## 0.02439024 0.02777778 0.03030303 0.02631579 0.02631579 0.02380952
```

```
betweenness(SMENetwork, directed=TRUE, weights = NA)
```

```
##           A           S           D           F           E           R           G
##  1.3005434  8.7400682  8.4077597  26.2159434  9.6454008  26.0980151  13.8489753
##           V           B           N           U           H           J           C
##  4.5213026  0.9154637  3.1412153  25.1877259  3.8342740  0.6036301  0.0000000
##           W           O           M           Q           T           I
##  0.0000000  0.5396825  0.0000000  0.0000000  0.0000000  0.0000000
```

- Appendix - Additional view

Clustering

```
par(mfrow=c(1,2))
par(mar=c(0.1,0.1,0.75,0.75))

par(bg="white") # set background

ceb <- cluster_edge_betweenness(SMENetwork)
dendPlot(ceb, mode="hclust") # plot(hcd, type = "triangle", ylab = "Height")
```

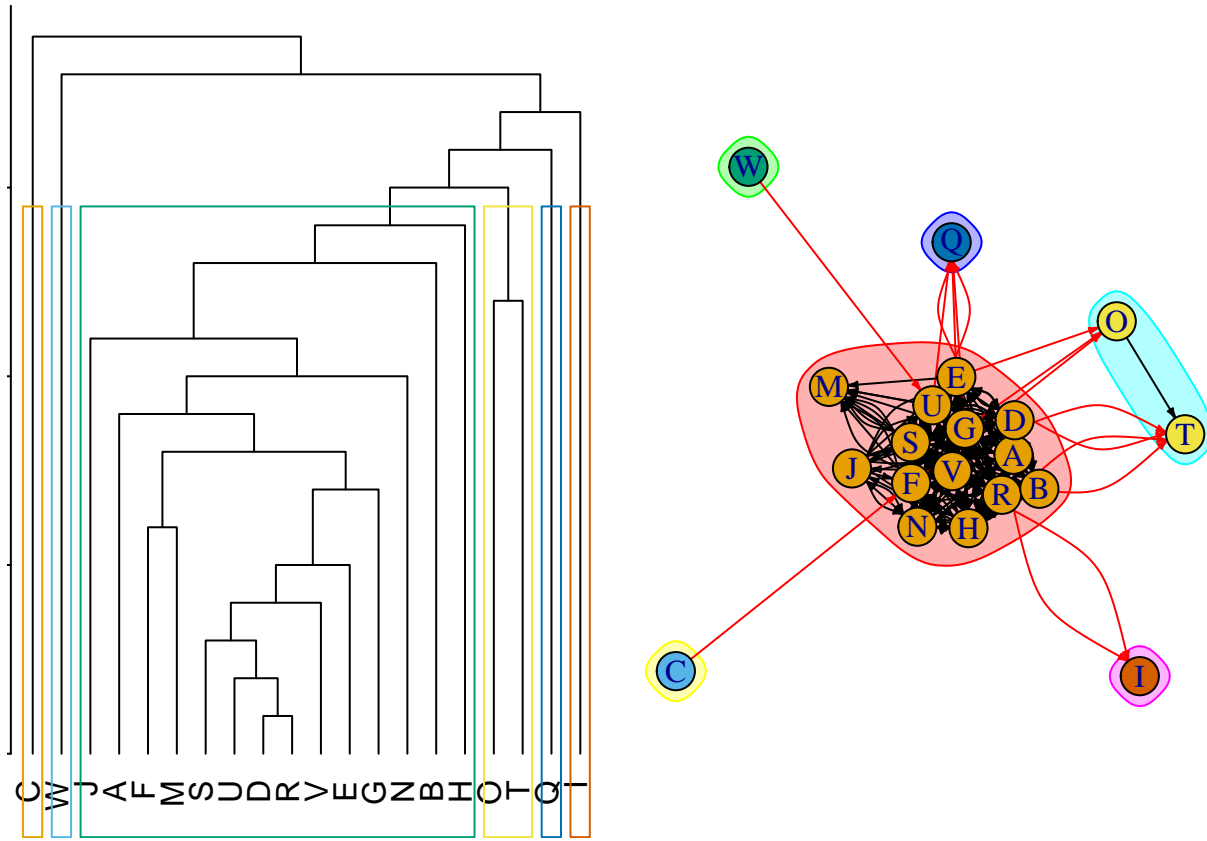
```
# greedy method (hiearchical, fast method)
coords = layout_with_fr(SMENetwork)

c3 = cluster_edge_betweenness(SMENetwork * 1.5)

# modularity measure
modularity(c3)
```

```
## [1] 0.006011823
```

```
# plot communities with shaded regions
plot_shaded_view <- plot(c3, SMENetwork, edge.arrow.size = 0.25, layout=coords); plot_shaded_view
```



NULL

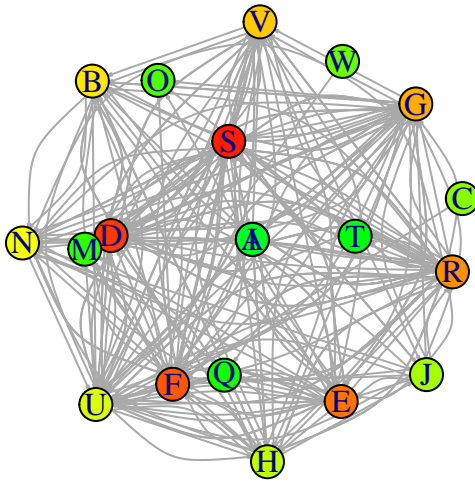
- comments: 6 outliers identified for dendrogram (outside green box) and network diagram (outside red area)

Additional network views

- Other views

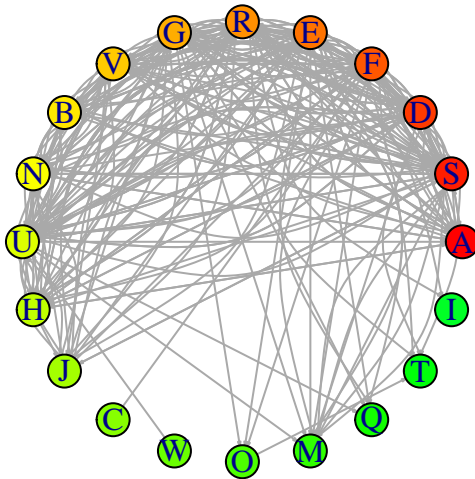
Spherical view

```
plot(SMEnetwork,
     vertex.color = rainbow(53),
     # vertex.size = V(SMEnetwork)$degree*0.4,
     edge.arrow.size = 0.1,
     layout=layout.sphere)
```



Circle view

```
plot(SMEnetwork,
     vertex.color = rainbow(53),
     # vertex.size = V(SMEnetwork)$degree*0.4,
     edge.arrow.size = 0.1,
     layout=layout.circle)
```

Star view

```
plot(SMENetwork,
     vertex.color = rainbow(56),
     # vertex.size = V(SMENetwork)$degree*0.4,
     edge.arrow.size = 0.1,
     layout=layout.star)
```

