

An Introduction to Network Analysis in R

Vinayak Ganesan

11/18/2020

Introduction

Throughout the course of this semester our research group has learned how to apply preliminary statistical analysis to biological networks using R. Primarily, we completed a brief literature review and identified the paper “A network biology-based approach to evaluating the effect of environmental contaminants on human interactome and diseases” by M. Lida et.Al. as the paper to replicate. Though the paper covered material beyond the scope of our project it presented us with the preliminary material and data to begin learning. Particularly, this paper provided us with the information to apply the stats and igraph packages in R as well as the original data obtained by the researchers. That said, in terms of replicating the paper we have created a Contaminant-Gene interaction network and ran a basic statistical analysis, producing similar if not identical results. The statistical analysis includes: a degree centrality analysis, closeness centrality analysis, betweenness centrality analysis, as well as the identification of clustering coefficients. Additionally to build on the work of the paper we have also applied the graphing analysis of “Inferring Virus-Host relationship between HPV and its host Homo sapiens using protein interaction network” by Q. Farooq et.Al developing both a degree Vs closeness and degree Vs betweenness plot, showcasing significant elements of the Contaminant-Gene network. That said, As we have now developed the methodology to properly explore a network using R, we intend to apply this knowledge to other datasets in the future in effort to gain better visual and statistical insight into biological networks.

Setup

Loads in libraries

```
library(readxl)
library(igraph)
library(data.table)
library(ggplot2)
library(writexl)
library(ggrepel)
```

Defines negate function

```
'%ni%' <- Negate('%in%')
```

Data Preperation

Sets wd and imports excel file and coerces into dataframe

```
setwd("C:/R/Replicating of A network biology-based approach to evaluating the effect of
environmental contaminants on human interactome and diseases")
data1 <- read_excel("Contaminant_Gene_Data.xlsx")
```

Creates a list of all the genes and chemicals

```
df1 <- data.frame(data1[,c(1,2,5,6,7,8,9,10,11,12)])
```

Labels all values as one of seven categories based on matrix and generates a unique df that contains only the Chemical and its respective label

```
for(i in 1:nrow(df1)){
  if (df1[i,3] == 1){
    df1[i,11] <- 2 #2 represents POPs
  }else if (df1[i,4] == 1){
    df1[i,11] <- 3 #3 represents Dioxins
  }else if (df1[i,5] == 1){
    df1[i,11] <- 4 #4 represents PAHs
  }else if (df1[i,6] == 1){
    df1[i,11] <- 5 #5 represents Metals
  }else if (df1[i,7] == 1){
    df1[i,11] <- 6 #6 Represents Pesticides
  }else if (df1[i,8] == 1){
    df1[i,11] <- 7 #7 represents PFCs
  }else if (df1[i,9] == 1){
    df1[i,11] <- 8 #8 represents PPCPs
  }else if (df1[i,10] == 1){
    df1[i,11] <- 9 #9 represents app_drug
  }
}
```

Creates a dataframe of Chemicals and their respective classification

```
chemicalLabel <- unique(cbind(df1[,2],df1[,11]))
colnames(chemicalLabel) <- c("Chemical Name","Chemical Classification")
```

Creates a unique Node df

```
nodes <- data.frame(unique(cbind(c(df1[,1],df1[,2]))))
colnames(nodes) <- c("Nodes")
```

Creates a edge df

```
edges <- data.frame(cbind(df1[,1],df1[,2]))
colnames(edges) <- c("Genes","Chemicals")
```

Identifies whether a node is a gene or chemical labels by 1 or 2 respectively and adds it to a new column 2 in nodes which will be named Type

```

for (i in 1:nrow(nodes)){
  if (nodes[i,1] %in% df1[,1]){
    nodes[i,2] <- 1 # Genes are 1

  }else{
    nodes[i,2] <- 2 #Contaminants are 2
  }
}

```

Identifies a node label as gene or chemical label and adds it to a new column 3 in nodes which will be named Label

```

for (i in 1:nrow(nodes)){
  if (nodes[i,2] == 1){
    nodes[i,3] <- 1 # Genes are 1

  }else{
    for(j in 1:nrow(chemicalLabel)){
      if(nodes[i,1] == chemicalLabel[j,1]){
        nodes[i,3] <- chemicalLabel[j,2]
      }
    }
  }
}

```

Finalizes nodes and adds labels

```

colnames(nodes) <- c("Nodes","Type","Label")
nodes$Label <- as.numeric(as.character(nodes$Label))

```

Creates a net and simplifies it to remove loops

```

net <- graph_from_data_frame(v= nodes, d=edges ,directed = T)
net <- simplify(net, remove.loops = T)

```

Restricting the degree, producing nodes only with degree greater than 70

```

nodeDegree <- degree(net, mode="all")
Degree69PlusBool <- nodeDegree > 69
NodeDegree69 <- nodeDegree[Degree69PlusBool]
nodes69 <- nodes[Degree69PlusBool,]

```

Restricting the degree, producing edges only with nodes with degree greater than 70

```

for(i in 1:nrow(edges)){
  if(edges[i,1] %ni% nodes69[,1]){
    edges[i,3] <- FALSE
  }else if(edges[i,2] %ni% nodes69[,1]){

```

```

        edges[i,3] <- FALSE
      }else{
        edges[i,3] <- TRUE
      }
    }

delEdges <- edges[,3]
str(delEdges)
edges69 <- edges[delEdges,]

```

Creating The Inital Network

Creates a node restricted net and simplifies it to remove loops

```

net <- graph_from_data_frame(v= nodes69, d=edges69 ,directed = T)
net <- simplify(net, remove.loops = T)

```

Creates color scheme and identifies the size of the nodes based on degree

```

nodeColors <- rainbow(9, alpha=.5)
V(net)$color <- nodeColors[V(net)$Label]
V(net)$size <- NodeDegree69*.05

```

plots the net

```

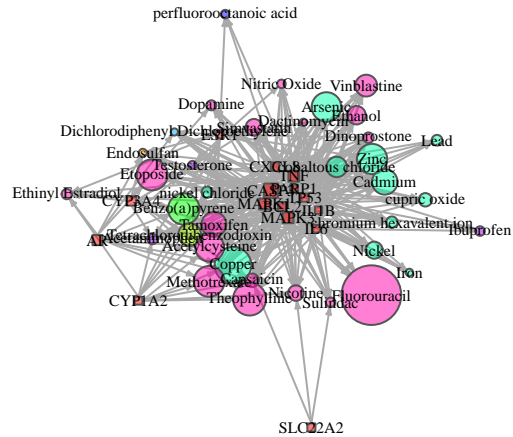
plot.igraph(net, vertex.shape = shapes()[c(3,1)][V(net)$Type], edge.arrow.size=.2,
            edge.curved=0,

            vertex.frame.color="#555555",

            vertex.label= V(net)$Nodes, vertex.label.color="black",

            vertex.label.cex=.5, layout = layout_fruchterman_reingold)

```



Basic Network Analysis

Degree Analysis

```

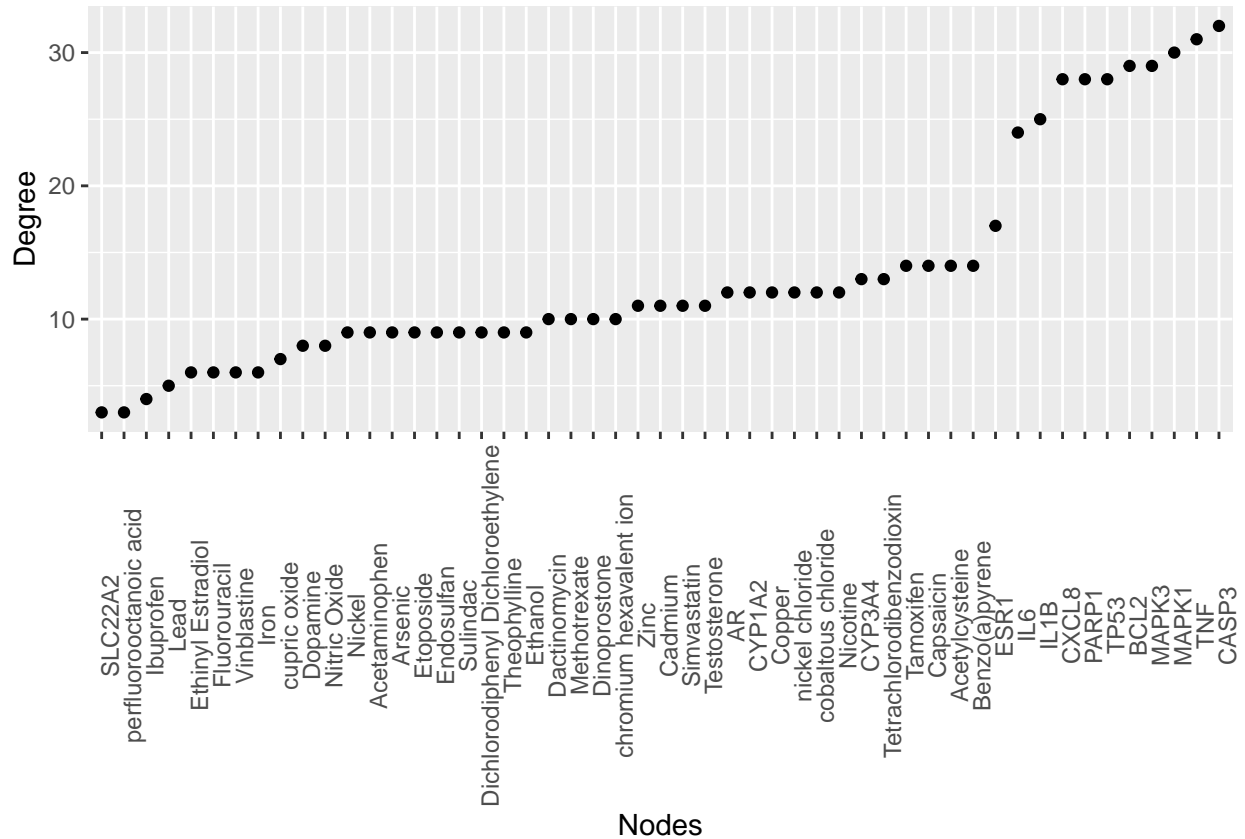
degree <- centr_degree(net, mode = c("total"), loops = FALSE, normalized = TRUE)
degree2 <- as.data.frame(setDT(as.data.frame(degree(net,mode = "all")), keep.rownames =
  TRUE) [])

for(i in 1:nrow(degree2)){ if(degree2[i,2] > 0){ degree2[i,3] <- TRUE }else{ degree2[i,3]
  <- FALSE }

}

delRowDegree <- degree2[,3]
degree3 <- degree2[delRowDegree,]
colnames(degree3)<- c("Nodes","Degree")
degree3$Nodes <- factor(degree3$Nodes, levels = degree3$Nodes[order(degree3$Degree)])
ggplot(degree3,aes(x= Nodes,y = Degree)) + geom_point() + theme(axis.text.x =
  element_text(angle = 90))

```



Betweenness Analysis

```

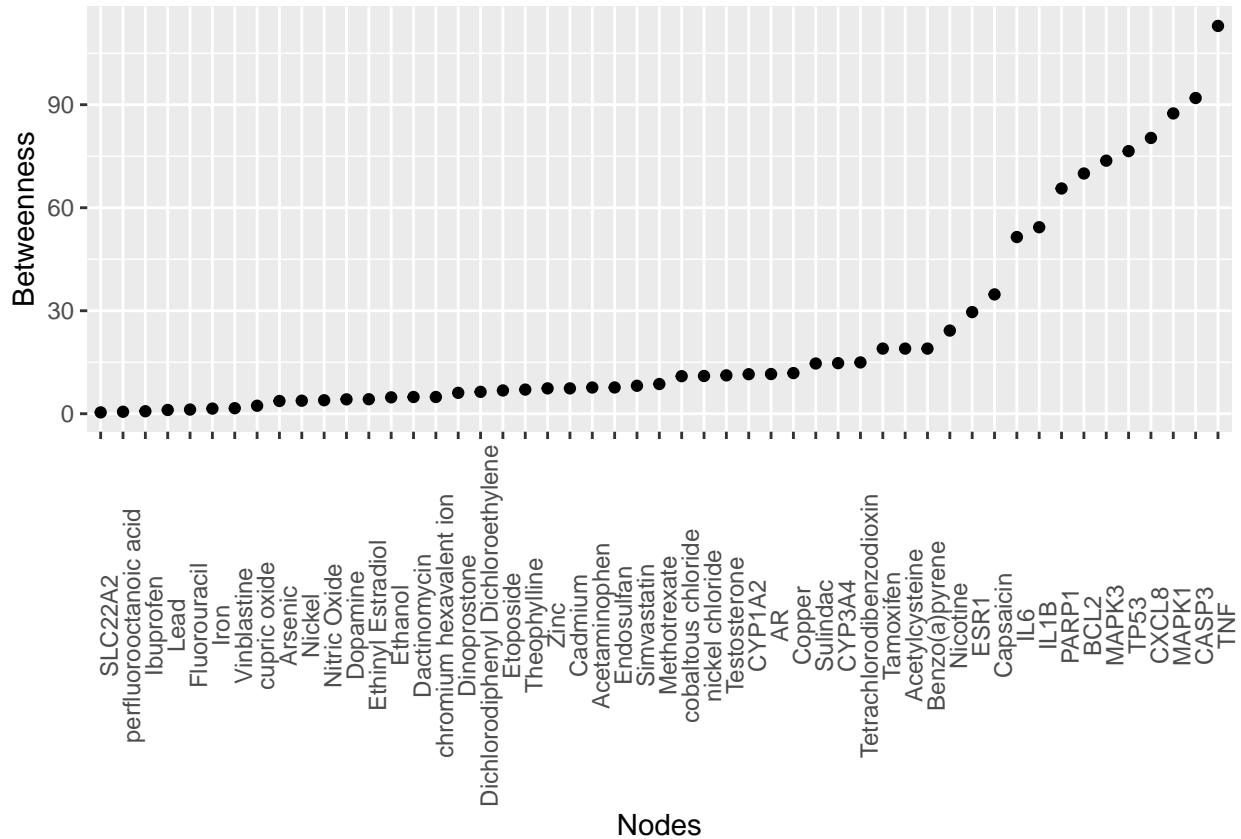
Betweenness <- betweenness(net, v = V(net), directed = FALSE, weights = NULL, nobigint =
  TRUE, normalized = FALSE)
Betweenness2 <- as.data.frame(setDT(as.data.frame(Betweenness), keep.rownames = TRUE)[])

for(i in 1:nrow(Betweenness2)){ if(Betweenness2[i,2] > 0){ Betweenness2[i,3] <- TRUE
  }else{ Betweenness2[i,3] <- FALSE }

}

delRowBetweenness <- Betweenness2[,3]
Betweenness3 <- Betweenness2[delRowBetweenness,]
colnames(Betweenness3)<- c("Nodes","Betweenness")
Betweenness3$Nodes <- factor(Betweenness3$Nodes, levels =
  Betweenness3$Nodes[order(Betweenness3$Betweenness)])
ggplot(Betweenness3,aes(x= Nodes,y = Betweenness)) + geom_point() + theme(axis.text.x =
  element_text(angle = 90))

```



Closeness Analysis

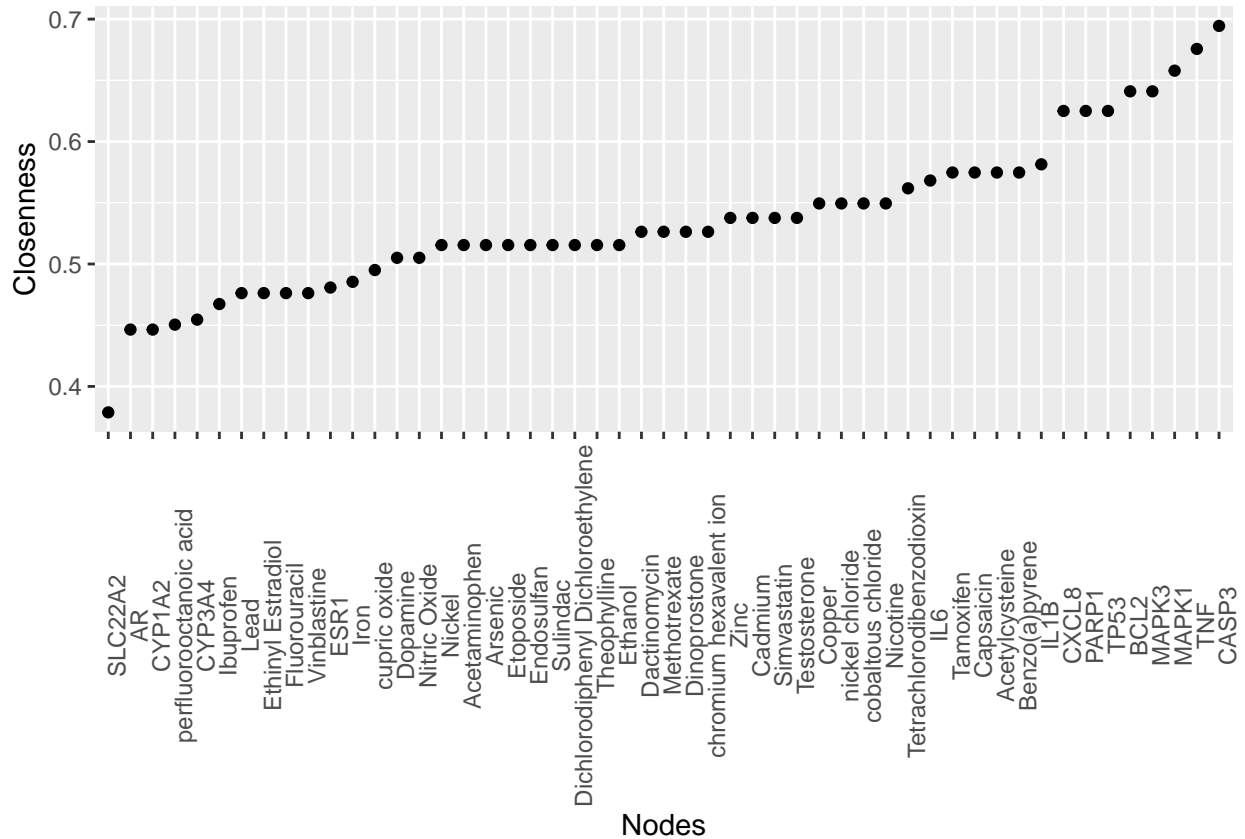
```

Closeness <- closeness(net, vids = V(net), mode = c("all"), weights = NULL, normalized =
  TRUE)
Closeness2 <- as.data.frame(setDT(as.data.frame(Closeness), keep.rownames = TRUE)[])

for(i in 1:nrow(Closeness2)){ if(Closeness2[i,2] > 0){ Closeness2[i,3] <- TRUE }else{
  Closeness2[i,3] <- FALSE }

}
delRowCloseness3 <- Closeness2[,3]
Closeness3 <- Closeness2[delRowCloseness3,]
colnames(Closeness3)<- c("Nodes", "Closeness")
Closeness3$Nodes <- factor(Closeness3$Nodes, levels =
  Closeness3$Nodes[order(Closeness3$Closeness)])
ggplot(Closeness3,aes(x= Nodes,y = Closeness)) + geom_point() + theme(axis.text.x =
  element_text(angle = 90))

```



Clustering Coefficient Analysis

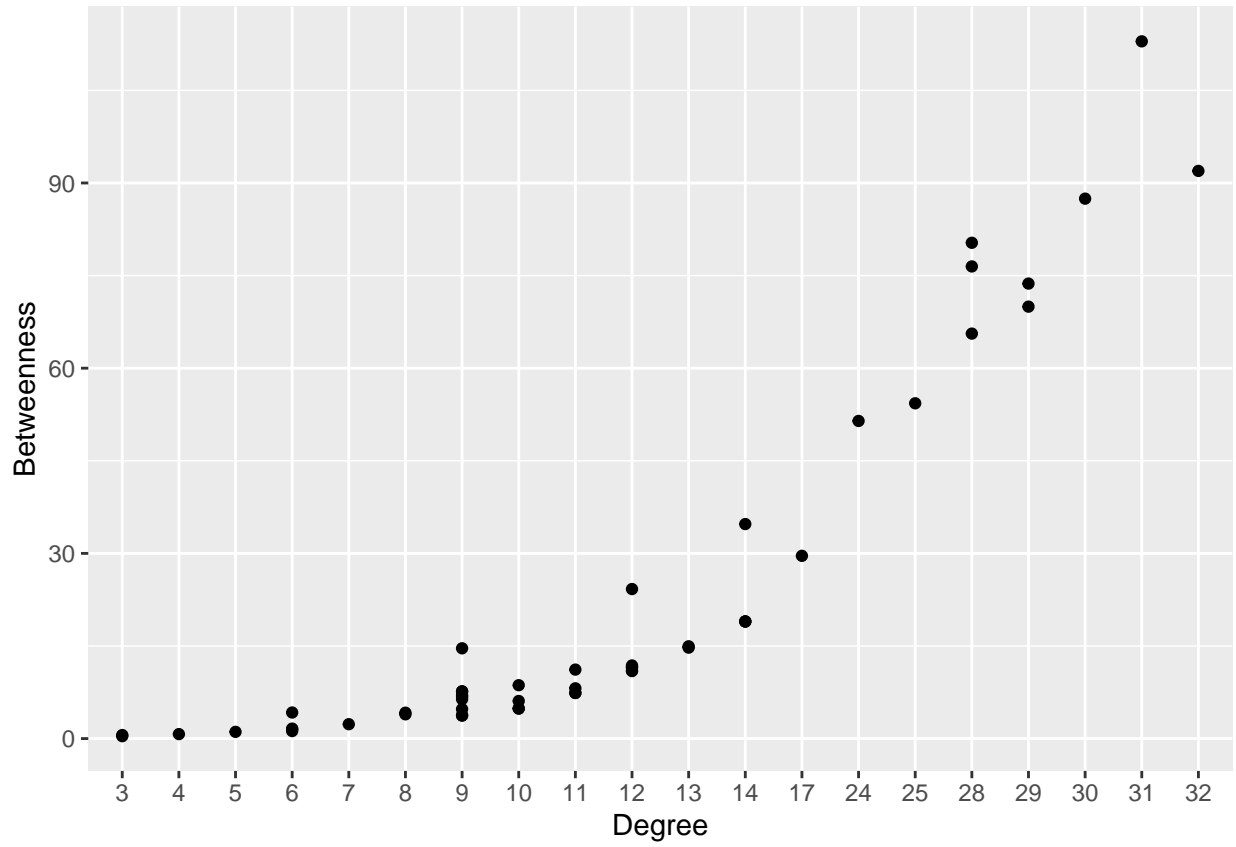
```
Transitivity <- transitivity(net, type = c("local"), vids = V(net), weights = NULL,
  isolates = c("NaN", "zero"))
```

Graphing Dual Relationships

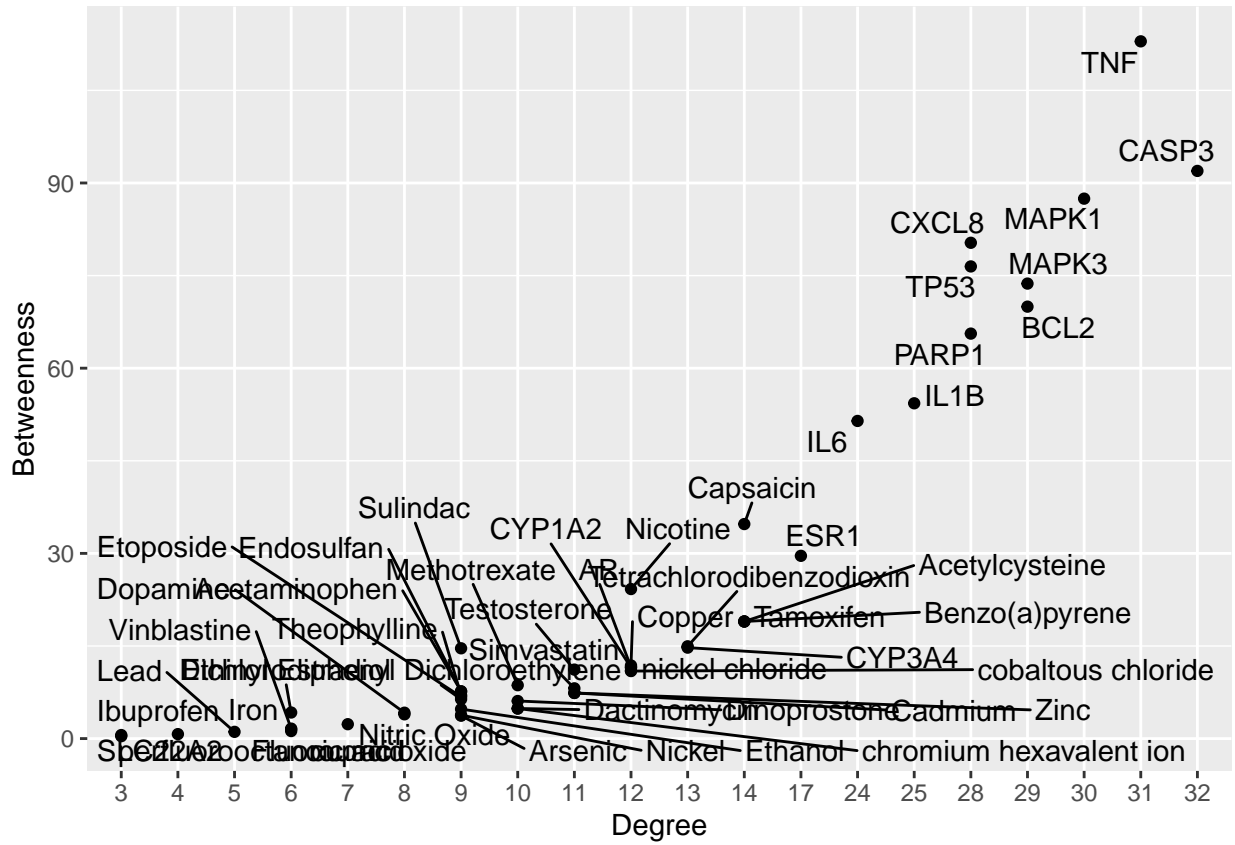
Degree and Betweenness

```
degreeBetweenness <- as.data.frame(cbind(degree2[,1], degree2[,2], Betweenness2[,2]))
degreeBetweenness[,2] <- as.double(degreeBetweenness[,2])
degreeBetweenness[,3] <- as.double(degreeBetweenness[,3])
colnames(degreeBetweenness) <- c("Node", "Degree", "Betweenness")

degreeBetweenness$Degree <- factor(degreeBetweenness$Degree, levels =
  sort(unique(degreeBetweenness$Degree[order(degreeBetweenness$Betweenness)])))
ggplot(degreeBetweenness, aes(x= Degree, y = Betweenness, label=degreeBetweenness[,1])) +
  geom_point()
```

```
ggplot(degreeBetweenness, aes(x= Degree, y = Betweenness, label=degreeBetweenness[,1])) +
  geom_point() + geom_text_repel()
```



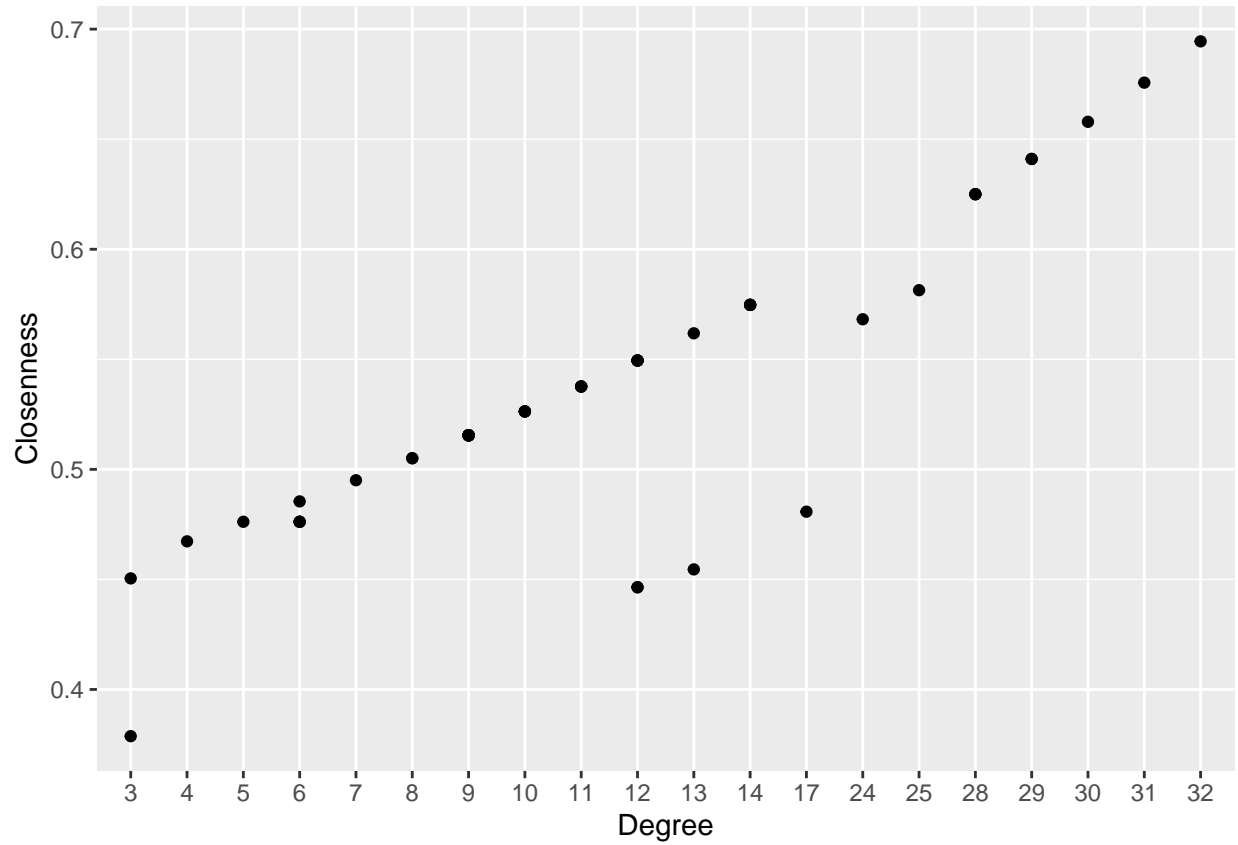
Degree and Closeness

```

degreeCloseness <- as.data.frame(cbind(degree2[,1],degree2[,2],Closeness2[,2]))
degreeCloseness[,2] <- as.double(degreeCloseness[,2])
degreeCloseness[,3] <- as.double(degreeCloseness[,3])
colnames(degreeCloseness)<- c("Node","Degree","Closeness")

degreeCloseness$Degree <- factor(degreeCloseness$Degree, levels =
  sort(unique(degreeCloseness$Degree[order(degreeCloseness$Closeness)])))
ggplot(degreeCloseness,aes(x= Degree,y = Closeness, label=degreeCloseness[,1])) +
  geom_point()

```



```
ggplot(degreeCloseness, aes(x= Degree, y = Closeness, label=degreeCloseness[,1])) +  
  geom_point() + geom_text_repel()
```

