Interactomic_Practise

Álvaro Ponce Cabrera

April 17, 2016

1. Recover the interactions corresponding to your pathogen from the IntAct database in a two-column text file

```
library(igraph)
```

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

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

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

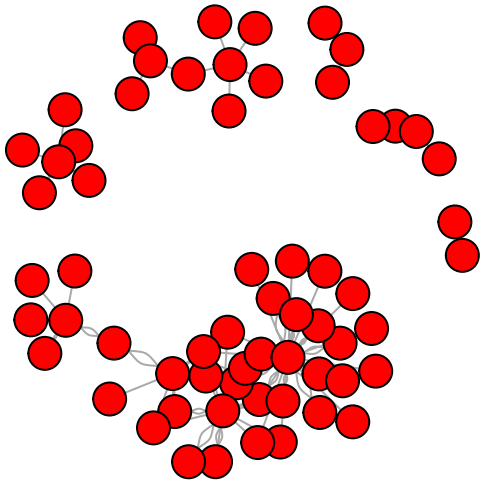
```
library(data.table)

setwd("C:/Users/alvaro/Desktop")

#Read the intact.txt file

intact<- fread('intact.txt', header =TRUE, data.table = FALSE)
```

```
##
Read 0.0% of 591277 rows
Read 6.8% of 591277 rows
Read 11.8% of 591277 rows
Read 20.3% of 591277 rows
Read 23.7% of 591277 rows
Read 30.4% of 591277 rows
Read 38.9% of 591277 rows
Read 45.7% of 591277 rows
Read 52.4% of 591277 rows
Read 60.9% of 591277 rows
Read 64.3% of 591277 rows
Read 72.7% of 591277 rows
Read 81.2% of 591277 rows
Read 89.6% of 591277 rows
Read 98.1% of 591277 rows
Read 591277 rows and 42 (of 42) columns from 2.160 GB file in 00:00:27
```

```r
#Looking for the pathogen
species<-unique(intact[,10])
# species

#The pathogen selected is: Human adenovirus C serotype 5

#Looking for interactions Human-Pathogen (HP) & Pathogen-Human (PH)

HumanIntactPositions<- which(intact[,10] =="taxid:9606(human)|taxid:9606(Homo sapiens)" &
  intact[,11] =="taxid:28285(ade05)|taxid:28285(\"Human adenovirus C serotype 5 (HAdV-5)\")"  )


HumanIntactPositions2<- which(intact[,11] =="taxid:9606(human)|taxid:9606(Homo sapiens)" &
 intact[,10] =="taxid:28285(ade05)|taxid:28285(\"Human adenovirus C serotype 5 (HAdV-5)\")" )


#Interactions Human-Pathogen where human is in ID A
H_P <- intact[HumanIntactPositions,]

#Interactions Pathogen-Human where human is in ID B
P_H<- intact[HumanIntactPositions2,]


#To work we only want the Protein ID of the interactors

HP<-H_P[,c(1:2)]
PH<-P_H[,c(1:2)]

#Bind both data frames, we have in one data frame H-P & P-H interactions
HP_PH <- rbind(HP,PH)
```

2. Build the network and analyze it:

```r
#Building the graph from HP-PH interact data
x<-graph.data.frame(HP_PH,directed=F)

plot(x,vertex.label.color ='transparent',vertex.color='red',vertex.label.dist=1)
```



1. How many components there are?

```
#Components of a graph: each set of nodes that can be reached walking trough edges
#form a component

components(x)$no
```

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

2. What is the size of the different components?

```
components(x)$csize
```

```
## [1] 38  9  2  4  6  3
```

3. What is the degree distribution of the network?

```
degree(x)
```

```
##        uniprotkb:Q13363      uniprotkb:Q01860-1        uniprotkb:Q92830
##                       3                       2                       2
```

```
##        uniprotkb:Q9Y4A5        uniprotkb:Q92831        uniprotkb:O15151
##                       2                       6                       3
##        uniprotkb:Q15326        uniprotkb:Q09472        uniprotkb:P06400
##                       4                       7                      13
##      uniprotkb:P10826-2        uniprotkb:Q9Y463      uniprotkb:P29590-3
##                       2                       2                       4
##        uniprotkb:P17980        uniprotkb:Q9H204        uniprotkb:P62195
##                       4                       2                       4
##        uniprotkb:P20718        uniprotkb:P06748        uniprotkb:Q9UER7
##                       1                       9                       4
##        uniprotkb:P08709        uniprotkb:Q92793        uniprotkb:P28749
##                       1                       4                       3
##        uniprotkb:P78396        uniprotkb:P21675        uniprotkb:P15927
##                       1                       3                       2
## ensembl:ENSG00000145386 ensembl:ENSG00000078900      uniprotkb:P29590-5
##                       1                       1                       2
##      uniprotkb:P29590-2      uniprotkb:P29590-1      uniprotkb:P29590-4
##                       2                       1                       1
##        uniprotkb:O60934      uniprotkb:P29590-8        uniprotkb:P29590
##                       1                       1                       4
##        uniprotkb:Q7Z7A1        uniprotkb:Q9Y2I6        uniprotkb:O15259
##                       2                       2                       1
##        uniprotkb:Q6UVJ0        uniprotkb:P03255      uniprotkb:P03255-2
##                       1                      49                      24
##      uniprotkb:P03255-1        uniprotkb:P03265        uniprotkb:P68951
##                      13                       5                       6
##        uniprotkb:P24938        uniprotkb:P03243        uniprotkb:P04133
##                       4                      10                       2
##      uniprotkb:P03243-1        uniprotkb:P04489        uniprotkb:P24933
##                       8                       6                       1
##        uniprotkb:P63244        uniprotkb:P42224        uniprotkb:P62826
##                       2                       4                       2
##        uniprotkb:Q13200        uniprotkb:P62333        uniprotkb:P10144
##                       1                       1                       1
##      uniprotkb:Q01105-2        uniprotkb:P00742        uniprotkb:Q08999
##                       1                       1                       2
##        uniprotkb:P20226        uniprotkb:Q8WXE1        uniprotkb:Q92547
##                       1                       1                       1
##        uniprotkb:Q13535        uniprotkb:Q01094
##                       1                       1
```

4. Which are the top ten human proteins with highest degree in the human-pathogen network?

```
#Degrees of the interactions
d<-degree(x)
head(d)
```

```
##   uniprotkb:Q13363 uniprotkb:Q01860-1   uniprotkb:Q92830
##                  3                  2                  2
##   uniprotkb:Q9Y4A5   uniprotkb:Q92831   uniprotkb:O15151
##                  2                  6                  3
```

```
length(d)
```

```
## [1] 62
```

```
#We use names(d) to find those nodes which are human
#proteins
#First we look for this names(d) into ID interactor A
#group

positions<-c()
for (i in 1:length(H_P[,1]))
{
  #Position of names(d) in ID interactor A group
  positions<-c(positions,grep(H_P[i,1],names(d)))
  #Save these names in d_human1 variable
  d_human1<-d[positions]
  #Eliminate repetitives names
  d_human1<- unique(names(d_human1))

  }

#Same as before but now with ID interactor B group
positions<-c()
for (i in 1:length(P_H[,2]))
{
  positions<-c(positions,grep(P_H[i,2],names(d)))
  positions
  d_human2<-d[positions]
  d_human2<- unique(names(d_human2))
}

#Merge of the two variables human proteins
d_human<- unique(c(d_human1,d_human2))
#In human we have all names(d) that correspond with
#human proteins

#Now we match these proteins with names(d) to have
#d positions where we have human proteins
positions<-c()
for (i in 1:length(d_human))
{
  positions<- c(positions, match(d_human[i],names(d)))
}
positions
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## [24] 24 25 26 27 28 29 30 31 32 33 34 35 36 37 49 50 51 52 53 54 55 56 57
## [47] 58 59 60 61 62
```

```
#Degrees from human proteins
d_h<-d[positions]
```

```
#Order of these degrees
d_h<-d_h[order(d_h,decreasing=TRUE)]

#Top ten human proteins with highest degree
d_h[1:10]
```

```
##   uniprotkb:P06400   uniprotkb:P06748   uniprotkb:Q09472
##                 13                  9                  7
##   uniprotkb:Q92831   uniprotkb:Q15326 uniprotkb:P29590-3
##                  6                  4                  4
##   uniprotkb:P17980   uniprotkb:P62195   uniprotkb:Q9UER7
##                  4                  4                  4
##   uniprotkb:Q92793
##                  4
```

5. Which are the top ten pathogen proteins with highest degree in the human-pathogen network?

```
#Same as 4, but with pathogen proteins
d<-degree(x)
head(d)
```

```
##   uniprotkb:Q13363 uniprotkb:Q01860-1   uniprotkb:Q92830
##                  3                  2                  2
##   uniprotkb:Q9Y4A5   uniprotkb:Q92831   uniprotkb:O15151
##                  2                  6                  3
```

```
length(d)
```

```
## [1] 62
```

```
positions<-c()
for (i in 1:length(H_P[,2]))
{
  positions<-c(positions,grep(H_P[i,2],names(d)))
  positions
  d_pathogen1<-d[positions]
  d_pathogen1<- unique(names(d_pathogen1))

}
d_pathogen1
```

```
##  [1] "uniprotkb:P03255"   "uniprotkb:P03255-2" "uniprotkb:P03255-1"
##  [4] "uniprotkb:P04489"   "uniprotkb:P24933"   "uniprotkb:P68951"
##  [7] "uniprotkb:P24938"   "uniprotkb:P03243"   "uniprotkb:P03243-1"
## [10] "uniprotkb:P04133"   "uniprotkb:P03265"
```

```
positions<-c()
for (i in 1:length(P_H[,1]))
{
  positions<-c(positions,grep(P_H[i,1],names(d)))
  positions
  d_pathogen2<-d[positions]
  d_pathogen2<- unique(names(d_pathogen2))
}

d_pathogen<- unique(c(d_pathogen1,d_pathogen2))


positions<-c()
for (i in 1:length(d_pathogen))
{
  positions<- c(positions, match(d_pathogen[i],names(d)))
}

d_p<-d[positions]
d_p<-d_p[order(d_p,decreasing=TRUE)]

#top ten pathogen proteins
d_p[1:10]
```

```
##   uniprotkb:P03255 uniprotkb:P03255-2 uniprotkb:P03255-1
##                 49                  24                  13
##   uniprotkb:P03243 uniprotkb:P03243-1   uniprotkb:P04489
##                 10                   8                   6
##   uniprotkb:P68951   uniprotkb:P03265   uniprotkb:P24938
##                  6                   5                   4
##   uniprotkb:P04133
##                  2
```

3. Analyze the human proteins in the context of the human PPI network

```
#Creating human-human interaction network

H_H_IntactPositions<- which(intact[,10] =="taxid:9606(human)|taxid:9606(Homo sapiens)" &
                            intact[,11] =="taxid:9606(human)|taxid:9606(Homo sapiens)")

#Interactions human-human
H_H_Net <- intact[H_H_IntactPositions,]

#Human-Human interaction Data Frame only with proteins
#ID
HH<-H_H_Net[,c(1:2)]

#Graph data
x_human<-graph.data.frame(HH,directed=F)
```

```
# plot(x_human,vertex.label.color ='transparent',vertex.color='red',vertex.label.dist=1)

#Jpeg into zip file
```

1. Is it the centrality of human proteins interacting with pathogen proteins similar to those which does not interact with pathogen proteins? Use appropriate statistical methods to obtain statistical significance

To see the centrality we will focus in the degree of both networks, we will study it with the mean of these degree and also with a statistical test in order to check the statistical significance of the changes between the 2 data population.

```
#H-P degree
mean(degree(x))
```

```
## [1] 4.129032
```

```
#H-H degree
mean(degree(x_human))
```

```
## [1] 20.90972
```

```
#H-H network is bigger than H-P, so it's this is not strange

#Test if both distributions are normal or not
shapiro.test(degree(x))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  degree(x)
## W = 0.4451, p-value = 6.147e-14
```

```
#Shapiro.test only accept 5000 values as maximun, so we take 5000 random values from degree(x_human)
shapiro.test(degree(x_human)[runif(5000,min=1,max=length(degree(x_human)))])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  degree(x_human)[runif(5000, min = 1, max = length(degree(x_human)))]
## W = 0.33486, p-value < 2.2e-16
```

```
#both are not normal distributions, so we apply the wilcox.test

wilcox.test(degree(x),degree(x_human))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  degree(x) and degree(x_human)
## W = 337930, p-value = 2.921e-08
## alternative hypothesis: true location shift is not equal to 0
```

```
#The differences are not random
```

2. Are the human proteins interacting with pathogen proteins closer to each other than those which does not interact with pathogens?

Like with centrality, we can study this with a mean,in this case with the mean of closeness.

```
mean(closeness(x, mode='all'))
```

```
## [1] 0.0004949122
```

```
Closeness_human<-closeness(x_human,mode='all')
mean(Closeness_human)
```

```
## [1] 3.167207e-07
```

```
#Now, the differences are due to the structure of the network and not at all because
#of the size.
#H-H network is a very compact network, so this
#little mean in closeness is normal.
#In contrast, H-P network has 6 components
#differentiated between them, and each comoponent
#is more or less compact

#Again we can see the statistical significance
#checking before if the data follow or not
#a normal distribution

shapiro.test(closeness(x, mode='all'))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  closeness(x, mode = "all")
## W = 0.67218, p-value = 1.727e-10
```

```
shapiro.test(Closeness_human[runif(5000,min=1,max=length(Closeness_human))])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Closeness_human[runif(5000, min = 1, max = length(Closeness_human))]
## W = 0.074852, p-value < 2.2e-16
```

```r
#No normal distribution

wilcox.test(closeness(x,mode='all'),Closeness_human)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  closeness(x, mode = "all") and Closeness_human
## W = 1136500, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```r
#The differences of the data are not random
```

4. Make any further analysis that you find relevant

Due to the differences between the two networks, there are no more interesting anaysis to do.

5. Draw some conclusions and write a short report with the results and conclusions. At the end of the report, as complementary material, add the scripts that you have used.

As we could see, the networks are pretty different between them, there are no remarkable relation shown by the initial analysis, and the fact that some of the human proteins of the H-P network have only interactions with the pathogen, but no with other human proteins, can explain this.