Thanos Vs Tony Stark - Final Network Project

Nancy Castillo

July 21, 2018

Background

The purpose of this network visualization is to find the main differences between the inner Thanos and Tony Stark's inner networks, comparing network sizes, centralization and interconnection measures as well as comparing the number of cores present on each network.

Loading Packages

```
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(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:igraph':
##
##     as_data_frame, groups, union
```

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

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Data

This exercise will use the **Hero Marvel Network Data** available on http://syntagmatic.github.io/exposedata/marvel/Data

The **Hero Marvel Network Data** contains 2 columns: "hero1" and "hero2" and 574467 observations. There are some duplicates among the observations signaling more than one interaction between heroes.

For the purpose of this exercise, we will simplify the network by assigning different weights depending on the number of interactions among our heroes, we will also segment the networks based on the vertices "THANOS" and "TONY STARK"

```
hero  <- read.csv("hero-network.csv")
head(hero)
```

```
##                  hero1                hero2
## 1        LITTLE, ABNER       PRINCESS ZANDA
## 2        LITTLE, ABNER BLACK PANTHER/T'CHAL
## 3 BLACK PANTHER/T'CHAL       PRINCESS ZANDA
## 4        LITTLE, ABNER       PRINCESS ZANDA
## 5        LITTLE, ABNER BLACK PANTHER/T'CHAL
## 6 BLACK PANTHER/T'CHAL       PRINCESS ZANDA
```

Exploring Data

```
nrow(hero)
```

```
## [1] 574467
```

```
nrow(unique(hero[,c("hero1", "hero2")]))
```

```
## [1] 224181
```

Assigning Weights and Simplifying Network

```
hero$weight <- 1
hero <- aggregate(hero[,3], hero[,-3], sum)
hero <- hero[order(hero$hero1, hero$hero2),]

colnames(hero)[3] <- "weight"
rownames(hero) <- NULL


thanos <- hero %>%
  filter(grepl("THANOS", hero1)| grepl("THANOS", hero2))
```

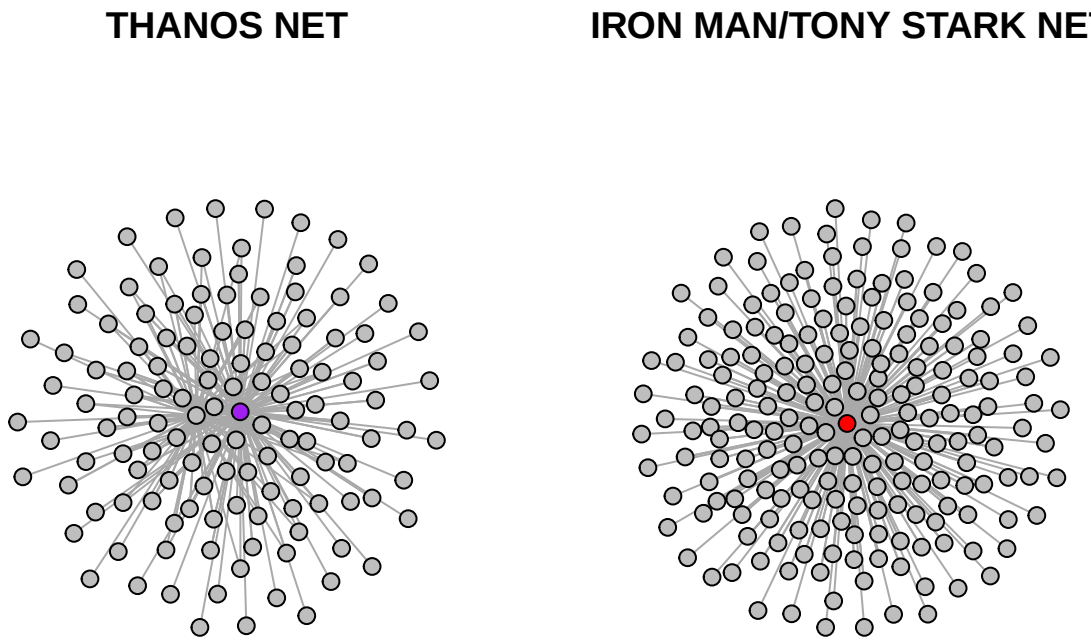
```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

```
cut.off <- mean(thanos$weight)


thanos <- thanos %>%
  filter(weight > cut.off)


tony <- hero %>%
  filter(grepl("TONY STARK", hero1)| grepl("TONY STARK", hero2))

cut.off <- mean(tony$weight)

tony <- tony %>%
  filter(weight > cut.off)
```

Creating Networks

For purposes of clarity for the visualization, **THANOS** will be assigned the "purple" color within his network and **Tony Stark** will be assigned the color "red"

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

## Thanos

thanos_n <- graph_from_data_frame(thanos, directed = T)
thanos_n <- simplify(thanos_n, remove.multiple = F, remove.loops = T)

vcol <- rep("gray", vcount(thanos_n))
vcol[V(thanos_n)$name == "THANOS"] <- "purple"

plot(thanos_n,
     vertex.color = vcol,
     vertex.size  = 8,
     edge.arrow.size  = .01,
     vertex.label = NA,
     layout = layout_with_lgl,
     main   = "THANOS NET")

## Tony

tony_n <- graph_from_data_frame(tony, directed = T)
tony_n <- simplify(tony_n, remove.multiple = F, remove.loops = T)


vcol <- rep("gray", vcount(tony_n))
vcol[V(tony_n)$name == "IRON MAN/TONY STARK "] <- "red"

plot(tony_n,
     vertex.color = vcol,
     vertex.size  = 8,
     edge.arrow.size  = .01,
     vertex.label = NA,
     layout = layout_with_lgl,
     main   = "IRON MAN/TONY STARK NET")
```



It seems like the Tony Stark's Network is more interconnected, we will verify this information below:

Comparing the two Networks

Who has the largest Network?

In terms of size, Tony's network is 76 actors larger than Thanos' Network

```
size_thanos <- length(V(thanos_n))
size_tony <- length(V(tony_n))

size <- matrix(c(size_thanos, size_tony), nrow = 1, ncol = 2)
colnames(size) <- c("Size Thanos", "Size Tony")

size
```

```
##      Size Thanos Size Tony
## [1,]         128       204
```

Network Centralization Measures

```
degree_centralization_thanos_n <- centralization.degree(thanos_n)$centralization
degree_sd_thanos_n             <- sd(centralization.degree(thanos_n)$res)

closeness_centralization_thanos_n<- centralization.closeness(thanos_n)$centralization
```

```
## Warning in centralization.closeness(thanos_n): At centrality.c:
## 2784 :closeness centrality is not well-defined for disconnected graphs
```

```
closeness_sd_thanos_n            <- sd(centralization.closeness(thanos_n)$res)
```

```
## Warning in centralization.closeness(thanos_n): At centrality.c:
## 2784 :closeness centrality is not well-defined for disconnected graphs
```

```
betweenness_centralization_thanos_n <- centralization.betweenness(thanos_n)$centralization
betweenness_sd_thanos_n            <- sd(centralization.betweenness(thanos_n)$res)


centralization_thanos_n <- matrix(c(degree_centralization_thanos_n,
                              degree_sd_thanos_n,
                              closeness_centralization_thanos_n,
                              closeness_sd_thanos_n,
                              betweenness_centralization_thanos_n,
                              betweenness_sd_thanos_n), nrow = 1, ncol = 6)

colnames(centralization_thanos_n) <- c("degree cent",
                                 "degree cent. SD",
                                 "closeness cent",
                                 "clossеness cent. SD",
                                 "betweenness cent",
                                 "betweenness cent SD")


degree_centralization_tony_n <- centralization.degree(tony_n)$centralization
degree_sd_tony_n             <- sd(centralization.degree(tony_n)$res)

closeness_centralization_tony_n<- centralization.closeness(tony_n)$centralization
```

```
## Warning in centralization.closeness(tony_n): At centrality.c:
## 2784 :closeness centrality is not well-defined for disconnected graphs
```

```
closeness_sd_tony_n            <- sd(centralization.closeness(tony_n)$res)
```

```
## Warning in centralization.closeness(tony_n): At centrality.c:
## 2784 :closeness centrality is not well-defined for disconnected graphs
```

```
betweenness_centralization_tony_n <- centralization.betweenness(tony_n)$centralization
betweenness_sd_tony_n            <- sd(centralization.betweenness(tony_n)$res)


centralization_tony_n <- matrix(c(degree_centralization_tony_n,
                              degree_sd_tony_n,
                              closeness_centralization_tony_n,
                              closeness_sd_tony_n,
                              betweenness_centralization_tony_n,
                              betweenness_sd_tony_n), nrow = 1, ncol = 6)

colnames(centralization_tony_n) <- c("degree cent",
                                 "degree cent. SD",
                                 "closeness cent",
                                 "clossеness cent. SD",
                                 "betweenness cent",
                                 "betweenness cent SD")
```

```
centralization <- rbind(centralization_thanos_n, centralization_tony_n)
rownames(centralization) <- c("Thanos", "Tony")

centralization_measures <- as.table(centralization)

centralization_measures
```

```
##         degree cent degree cent. SD closeness cent closseness cent. SD
## Thanos 7.236654e-01    1.891006e+01   4.162343e-03        7.895316e-03
## Tony   8.455677e-01    2.404021e+01   6.085331e-03        1.119636e-02
##        betweenness cent betweenness cent SD
## Thanos     4.988653e-01        7.360152e+02
## Tony       7.218944e-01        2.072555e+03
```

As expected, both Thano's and Tony's networks have a high degree centrality since there is an actor that place a central role in the network. This was expected as the result of how these networks were segmented, however, **degree centralization and betweenness centrality is highest for Tony's Network**; signaling that there may be another actor in Thano's network that plays a somehow central role

Who has the most interconnected Network?

```
density_thanos_n        <- edge_density(thanos_n)
average_degree_thanos_n <- mean(degree(thanos_n))
cohesion_thanos_n       <- cohesion(thanos_n)
compactness_fist        <- mean(closeness(thanos_n))
```

```
## Warning in closeness(thanos_n): At centrality.c:2617 :closeness centrality
## is not well-defined for disconnected graphs
```

```
global_clustering_coefficient_thanos_n <- transitivity(thanos_n, type = "global")

interconnectedness_thanos_n <- matrix(c(density_thanos_n,
                                        average_degree_thanos_n,
                                        cohesion_thanos_n,
                                        compactness_fist,
                                        global_clustering_coefficient_thanos_n),
                                        nrow = 1 , ncol = 5)

colnames(interconnectedness_thanos_n) <- c("density",
                                           "avg. degree",
                                           "cohesion",
                                           "compactness",
                                           "clustering coefficient")


density_tony_n        <- edge_density(tony_n)
average_degree_tony_n <- mean(degree(tony_n))
cohesion_tony_n       <- cohesion(tony_n)
compactness_fist      <- mean(closeness(tony_n))
```

```
## Warning in closeness(tony_n): At centrality.c:2617 :closeness centrality is
## not well-defined for disconnected graphs
```

```
global_clustering_coefficient_tony_n <- transitivity(tony_n, type = "global")

interconnectedness_tony_n <- matrix(c(density_tony_n,
                                      average_degree_tony_n,
                                      cohesion_tony_n,
                                      compactness_fist,
                                      global_clustering_coefficient_tony_n),
                                      nrow = 1 , ncol = 5)

colnames(interconnectedness_tony_n) <- c("density",
                                         "avg. degree",
                                         "cohesion",
                                         "compactness",
                                         "clustering coefficient")



interconnectedness <- rbind(interconnectedness_thanos_n, interconnectedness_tony_n)

rownames(interconnectedness) <- c("Thanos", "Tony")

interconnectedness <- as.table(interconnectedness)

interconnectedness
```

```
##             density  avg. degree     cohesion  compactness
## Thanos 1.820866e-02 4.625000e+00 0.000000e+00 1.733900e-04
## Tony   8.330919e-03 3.382353e+00 0.000000e+00 6.944978e-05
##        clustering coefficient
## Thanos           1.976732e-02
## Tony             0.000000e+00
```

For a large network **(204 actors)**,Tony's network is highly connected with a 8.33 density and **3.38 average degree** and it is the most **compacted network** However, Thano's network is also highly interconnected as shown in the plots

Who are some of the central actors within each network?

Finding the eigen centrality

Since we know Thanos and Tony are going to be "important" actors within their networks, we will use Eigenvalue Centrality to give more significance to those actors that are connected to other important actors in the network.

We will be using the Eigen Centrality scores as vertice size to help distinguish the actors that have the highest score.

Open the files: "thanos_n_gjs_ec.html" "tony_n_gjs_ec.html" to examin the networks

```
library("threejs")
```

```
## Warning: package 'threejs' was built under R version 3.3.3
```

```
library("htmlwidgets")
library("igraph")
```

```
thanos_ec <- eigen_centrality(thanos_n)$vector

tony_ec <- eigen_centrality(tony_n)$vector


## Eigenvalue Centrality Graphs

##Tony

net.js <- tony_n
graph_attr(net.js, "layout") <- NULL

# Create a vector of colors

colr <- rep("blue", vcount(net.js))

vcol <- rep("black", vcount(tony_n))
vcol[V(tony_n)$name == "IRON MAN/TONY STARK "] <- "brown2"

tony_ec <- eigen_centrality(tony_n)$vector

V(tony_n)$ec <- eigen_centrality(tony_n)$vector

l <- V(tony_n)$name

tony_n_gjs_ec <- graphjs(tony_n,
                         main="TONY STARK NET Eigenvalue Centrality",
                         showLabels = T,
                        vertex.label = l,
                        vertex.size = V(tony_n)$ec,
                        vertex.color = vcol,
                        edge.color = "gray",
                        attraction = 0.3,
                        repultion = 1,
                        opacy = 0.9)
print(tony_n_gjs_ec)
saveWidget(tony_n_gjs_ec, file="tony_n_gjs_ec.html")


net.js <- thanos_n
graph_attr(net.js, "layout") <- NULL

# Create a vector of colors

vcol <- rep("black", vcount(thanos_n))
vcol[V(thanos_n)$name == "THANOS"] <- "purple"

thanos_ec <- eigen_centrality(thanos_n)$vector

V(thanos_n)$ec <- eigen_centrality(thanos_n)$vector

l <- V(thanos_n)$name
```

```
thanos_n_gjs_ec <- graphjs(thanos_n,
                        main="THANOS Eigenvalue Centrality",
                        showLabels = T,
                       vertex.label = l,
                       vertex.size = V(thanos_n)$ec,
                       vertex.color = vcol,
                       edge.color = "gray",
                       attraction = 0.3,
                       repultion = 1,
                       opacy = 0.9)
print(thanos_n_gjs_ec)
saveWidget(thanos_n_gjs_ec, file="thanos_n_gjs_ec.html")
```

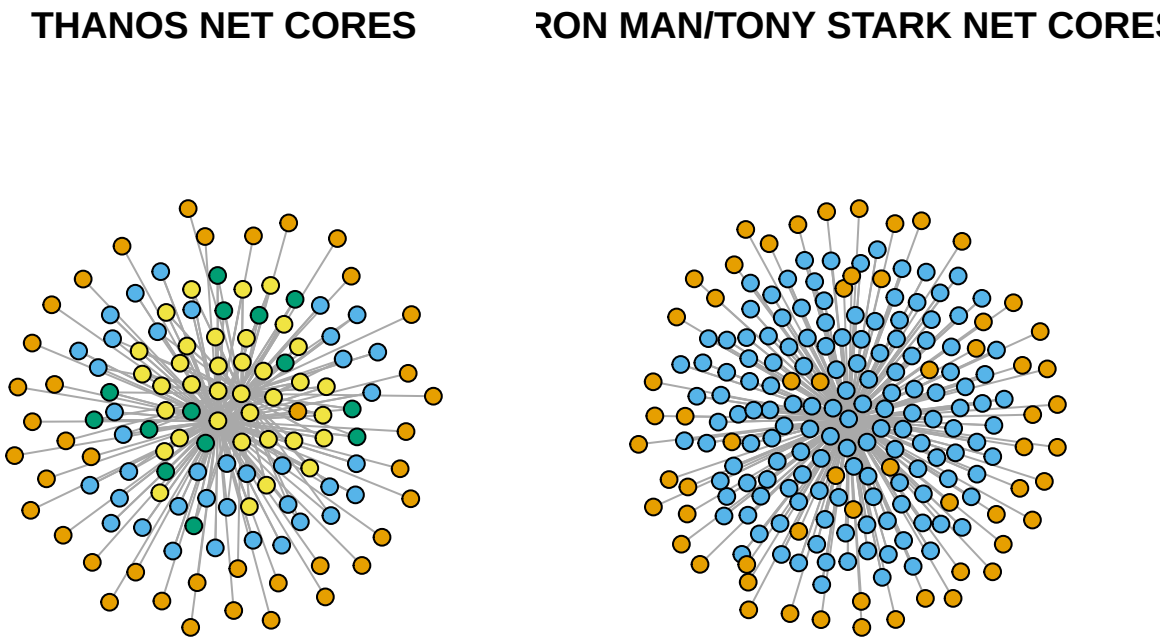
Within **Tony's Network**, **CAPITAN AMERICA,ANT MAN, SCARLET WITCH, THOR, VISION, AND WONDER MAN** have high Eigenvalue Centrality scores.

Within **Thanos' Network**, **THANOS Doppelanger** have the second highest Eigenvalue Centrality, making this actor the second most connected and influential actor within this network

Are there subgroups within these networks?

Thano's Network can be separated into 4 subgroups of actors while Tony's network appears to have 2 subgroups. The actors within these different subgroups are directed connected to and interact with each other at a higher rate. We are using colors to identify these cores and decided not to modify the size of the vertices to generate a cohesive graph/picture, additionally the layout **layout_with_lgl** is used since it showed to display the graph more clearly

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

thanos_n <- graph_from_data_frame(thanos, directed = T)
thanos_n <- simplify(thanos_n, remove.multiple = F, remove.loops = T)



plot(thanos_n,
     vertex.color = graph.coreness(thanos_n),
     vertex.size  = 8,
     edge.arrow.size  = .01,
     vertex.label = NA,
     layout = layout_with_lgl,
     main   = "THANOS NET CORES")

tony_n <- graph_from_data_frame(tony, directed = T)
tony_n <- simplify(tony_n, remove.multiple = F, remove.loops = T)


plot(tony_n,
     vertex.color = graph.coreness(tony_n),
     vertex.size  = 8,
     edge.arrow.size  = .01,
     vertex.label = NA,
     layout = layout_with_lgl,
     main   = "IRON MAN/TONY STARK NET CORES")
```



Putting it all toguether with graphjs

It is clear that the 2 core found within Tony's network has the lowest eigen centrality scores.

Opend the files "thanos_n_gjs_ec_coreness.html" & "tony_n_gjs_ec_coreness.html" to explore the networks. The size of the vertices represent the Eigen Value Centrality scores while the colors shoecase the different cores present within each network

```
## Eigenvalue Centrality Graphs

library("threejs")
library("htmlwidgets")
library("igraph")


##Tony

net.js <- tony_n
graph_attr(net.js, "layout") <- NULL

tony_ec <- eigen_centrality(tony_n)$vector

V(tony_n)$ec <- eigen_centrality(tony_n)$vector

V(tony_n)$coreness <- graph.coreness(tony_n)
```

```
vcol <- rep("blue", vcount(tony_n))

vcol[V(tony_n)$coreness == "1"] <-"green"
vcol[V(tony_n)$coreness == "2"] <-"brown2"


l <- V(tony_n)$name

tony_n_gjs_ec_coreness <- graphjs(tony_n,
                        main="TONY STARK NET Eigenvalue Centrality & Coreness",
                        showLabels = T,
                       vertex.label = l,
                       vertex.size = V(tony_n)$ec,
                       vertex.color = vcol,
                       edge.color = "gray",
                       attraction = 0.3,
                       repultion = 1,
                       opacy = 0.9)
print(tony_n_gjs_ec_coreness)
saveWidget(tony_n_gjs_ec_coreness, file="tony_n_gjs_ec_coreness2.html")


net.js <- thanos_n
graph_attr(net.js, "layout") <- NULL

# Create a vector of colors

V(thanos_n)$coreness <- graph.coreness(thanos_n)
vcol <- rep("purple", vcount(thanos_n))

vcol[V(thanos_n)$coreness == "1"] <-"red"
vcol[V(thanos_n)$coreness == "2"] <-"gold"
vcol[V(thanos_n)$coreness == "3"] <-"purple"
vcol[V(thanos_n)$coreness == "3"] <-"green"


thanos_ec <- eigen_centrality(thanos_n)$vector

V(thanos_n)$ec <- eigen_centrality(thanos_n)$vector

l <- V(thanos_n)$name

thanos_n_gjs_ec <- graphjs(thanos_n,
                        main="THANOS Eigenvalue Centrality & Coreness",
                        showLabels = T,
                       vertex.label = l,
                       vertex.size = V(thanos_n)$ec,
                       vertex.color = vcol,
                       edge.color = "gray",
                       attraction = 0.3,
                       repultion = 1,
                       opacy = 0.9)
print(thanos_n_gjs_ec)
```

```
saveWidget(thanos_n_gjs_ec, file="thanos_n_gjs_ec_coreness.html")
```

Conclusion

Both networks have a high degree centrality since the main actors (Thanos & Tony) place a central role in the network. However, **degree centralization and betweenness centrality is highest for Tony's Network** making this network **more interconnected**.

After analysing Thano's network, it was clear that the actor **"Thanos Doppelganger"** also played a central role within this network which affected its interconnection scores. Moreover, Thano's network also show to be more fragmentated with 4 cores. Tony's network only contained 2 cores and one of them showed to be highly interconnected