Analyzing centralitty measures for a sexual network of gonorrhea transmission

Abstract

Introduction

Methods

Results

Data Analysis

Centrality

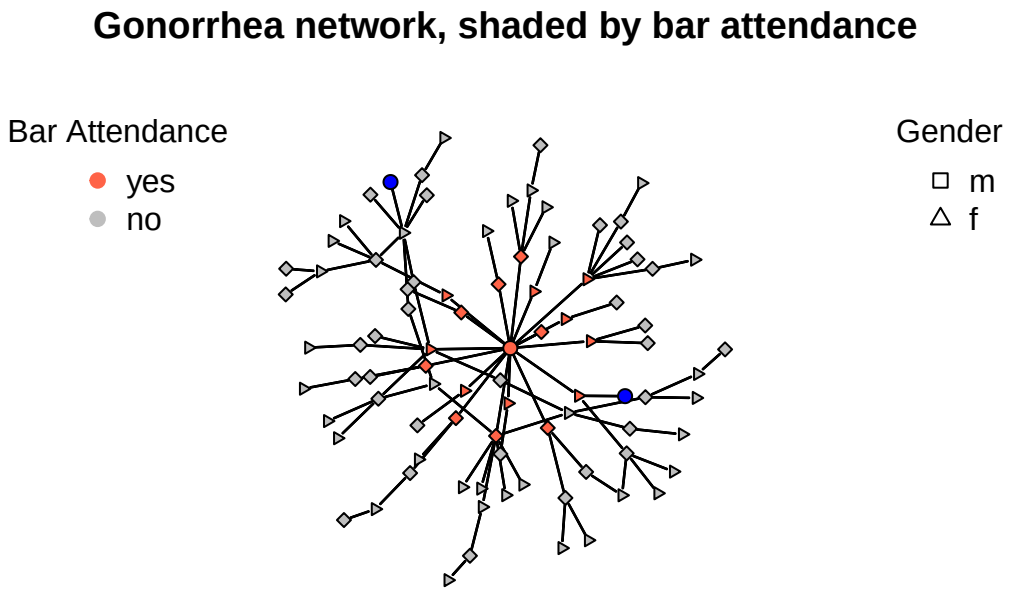
Network Density

K-cores

Community Detection

```
# data is loaded as 'gonnet'
nodes <- rownames(gonnet)
colnames(gonnet) <- nodes

# symmetrized graph (undirected)
gonnet_sym <- symmetrize(gonnet, rule = "weak")

gonnet_df <- data.frame(nodes) %>%
  mutate(gender = ifelse(grepl("m", nodes), yes = "m", no = NA)) %>%
  mutate(gender = ifelse(grepl("f", nodes), yes = "f", no = gender)) %>%
  mutate(gender_lty = ifelse(gender == "m", yes = 4, no = 3)) %>%
  mutate(gender_lty = ifelse(gender == "f", yes = 3, no = gender_lty)) %>%
  mutate(gender_lty = ifelse(gender %in% c("m", "f"), yes = gender_lty, no = 50)
         ) %>%
  mutate(attended_bar = gonnet["b",] == 1) %>%
  mutate(col = ifelse(attended_bar == T, "tomato1", "grey"))


gonnet_df$col[!(gonnet_df$gender %in% c("m", "f"))] <- "blue"
gonnet_df$col[gonnet_df$nodes == "b"] <- "tomato1"
```

```
par(mfrow = c(1,1))
set.seed(10)
org_coord <-
  gplot(gonnet_sym, vertex.col = gonnet_df$col,
      vertex.sides = gonnet_df$gender_lty,
      # displaylabels = T, label.cex = 0.6, label.pos = 2,
      boxed.labels = F, pad = 2, usearrows = F,
      vertex.cex = 1.25)
# legend for gender
legend("topleft",
       legend = c("yes", "no"),
       col = c("tomato1", "grey"),
       fill = F, border = "white", pch = 19,
       title = "Bar Attendance", bty = "n")
legend("topright",
       legend = c("m", "f"),
       col = c("black"),
       fill = F, border = "white", pch = c(0, 2),
       title = "Gender", bty = "n")
# legend for bar attendance
title("Gonorrhea network, shaded by bar attendance")
```




```
cen <- get_centralities(gonnet, "directed")

# correlations between centrality measures
```

```
centrality_correlations(gonnet, cen)
```

```
##              outdegree   indegree betweenness eigenvector
## outdegree    1.0000000 -0.1970015  0.40612528  0.79067530
## indegree    -0.1970015  1.0000000  0.52061632 -0.37578288
## betweenness  0.4061253  0.5206163  1.00000000 -0.05333083
## eigenvector  0.7906753 -0.3757829 -0.05333083  1.00000000
```

```
cen_sym <- get_centralities(gonnet_sym, "undirected")
centrality_correlations(gonnet_sym, cen_sym)
```

```
##              outdegree   indegree betweenness eigenvector
## outdegree    1.0000000  1.0000000   0.9543577  -0.8172166
## indegree     1.0000000  1.0000000   0.9543577  -0.8172166
## betweenness  0.9543577  0.9543577   1.0000000  -0.8603699
## eigenvector -0.8172166 -0.8172166  -0.8603699   1.0000000
```

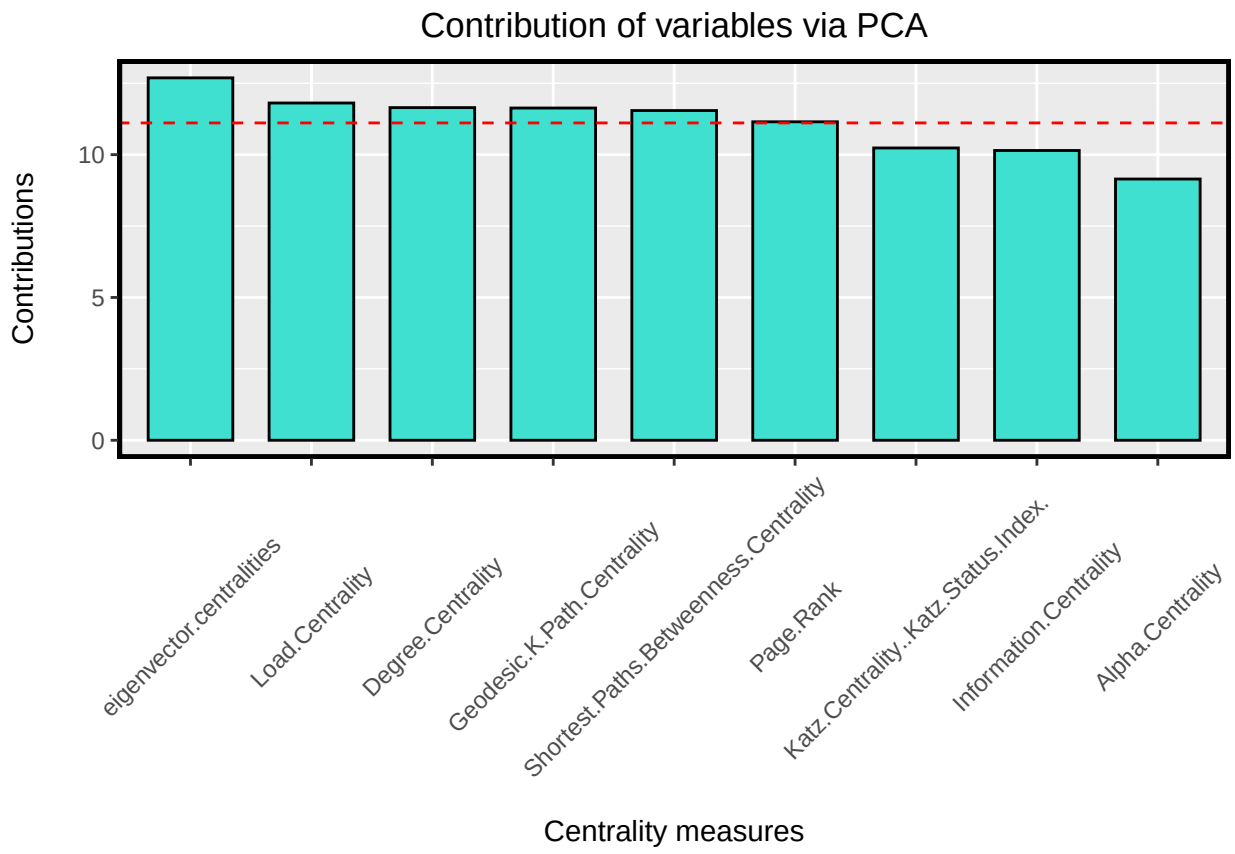
Notice that outdegree and indegree are negatively correlated, which makes sense in this context; an individual will only receive gonorrhea from one person (despite possibly having multiple sexual partners with a diagnosis)

```
# creating igraph object
gonnet_ig <- graph_from_adjacency_matrix(gonnet)

pc <- proper_centralities(gonnet_ig)
```

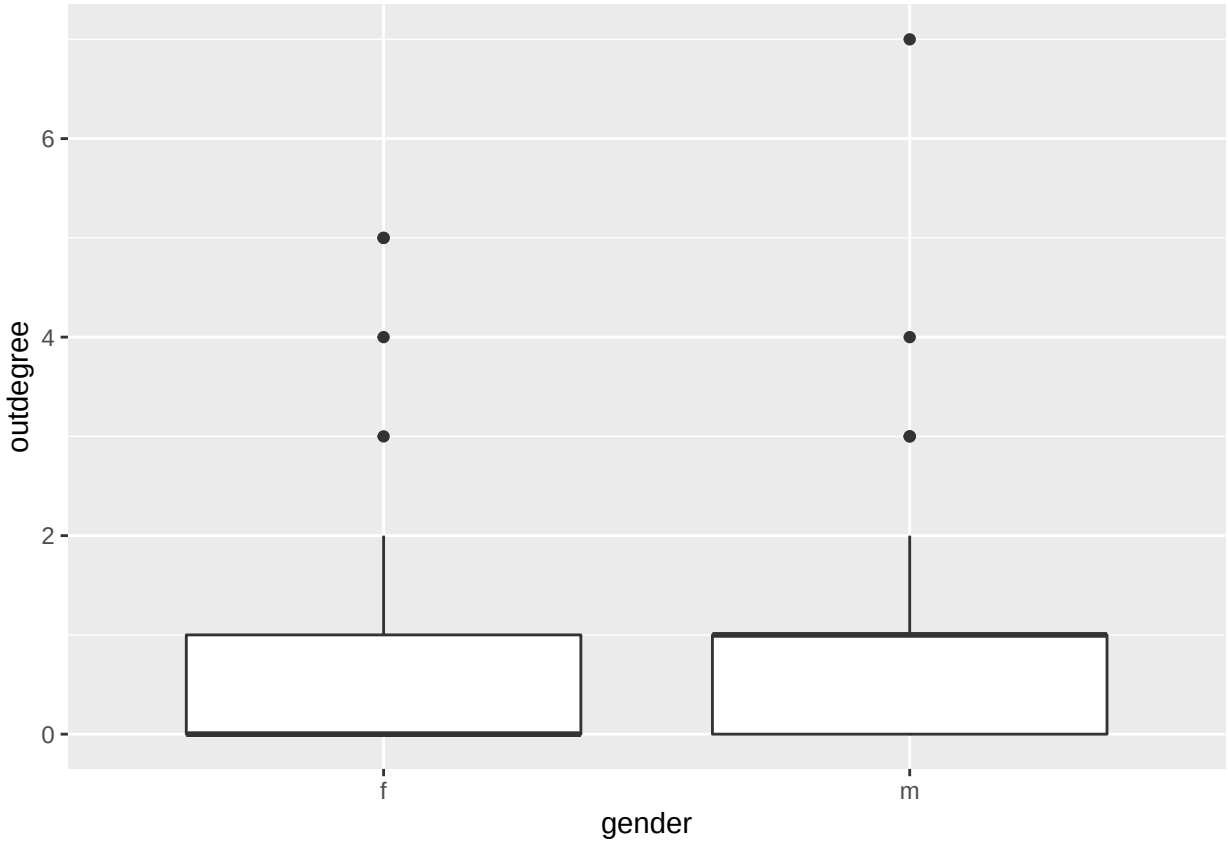
```
##  [1] "Alpha Centrality"
##  [2] "Bonacich power centralities of positions"
##  [3] "Page Rank"
##  [4] "Average Distance"
##  [5] "Barycenter Centrality"
##  [6] "BottleNeck Centrality"
##  [7] "Centroid value"
##  [8] "Closeness Centrality (Freeman)"
##  [9] "ClusterRank"
## [10] "Decay Centrality"
## [11] "Degree Centrality"
## [12] "Diffusion Degree"
## [13] "DMNC - Density of Maximum Neighborhood Component"
## [14] "Eccentricity Centrality"
## [15] "Harary Centrality"
## [16] "eigenvector centralities"
## [17] "K-core Decomposition"
## [18] "Geodesic K-Path Centrality"
## [19] "Katz Centrality (Katz Status Index)"
## [20] "Kleinberg's authority centrality scores"
## [21] "Kleinberg's hub centrality scores"
## [22] "clustering coefficient"
## [23] "Lin Centrality"
## [24] "Lobby Index (Centrality)"
## [25] "Markov Centrality"
```

```
## [26] "Radiality Centrality"
## [27] "Shortest-Paths Betweenness Centrality"
## [28] "Current-Flow Closeness Centrality"
## [29] "Closeness centrality (Latora)"
## [30] "Communicability Betweenness Centrality"
## [31] "Community Centrality"
## [32] "Cross-Clique Connectivity"
## [33] "Entropy Centrality"
## [34] "EPC - Edge Percolated Component"
## [35] "Laplacian Centrality"
## [36] "Leverage Centrality"
## [37] "MNC - Maximum Neighborhood Component"
## [38] "Hubbell Index"
## [39] "Semi Local Centrality"
## [40] "Closeness Vitality"
## [41] "Residual Closeness Centrality"
## [42] "Stress Centrality"
## [43] "Load Centrality"
## [44] "Flow Betweenness Centrality"
## [45] "Information Centrality"
## [46] "Dangalchev Closeness Centrality"
## [47] "Group Centrality"
## [48] "Harmonic Centrality"
## [49] "Local Bridging Centrality"
## [50] "Wiener Index Centrality"
```

```
centralities <- calculate_centralities(gonnet_ig,
                       include = pc[c(1, 4, 11, 16, 27,
                                      45, 8, 18, 31, 43, 3, 19)])

# plot of different centralities' contributions via PCA
pca_centralities(centralities)
```




```
gonnet_df2 <- cbind(gonnet_df, cen) %>% select(-id)

# boxplot comparing outdegree
outdegree_gender <- gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  drop_na()

ggplot(outdegree_gender, aes(x=gender, y=outdegree)) +
  geom_boxplot()
```




```
# average outdegree overall
gonnet_df2$outdegree %>% mean()
```

```
## [1] 1.044944
```

```
# average outdegree among males in network, overall
outdegree_m <- gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  filter(gender == "m") %>%
  select(outdegree) %>%
  unlist()
outdegree_m %>% mean()
```

```
## [1] 0.9767442
```

```
# male outdegree that excludes nodes with zero outdegree
gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  filter(gender == "m" & outdegree != 0) %>%
  select(outdegree) %>%
  unlist() %>% mean()
```

```
## [1] 1.826087
```

```r
# average outdegree among females
outdegree_f <- gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  filter(gender == "f") %>%
  select(outdegree) %>%
  unlist()
outdegree_f %>% mean()
```

```
## [1] 0.7906977
```

```r
# female outdegree that excludes nodes with zero outdegree
gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  filter(gender == "f" & outdegree != 0) %>%
  select(outdegree) %>%
  unlist() %>% mean()
```

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

```r
# levene test for equality of variances
# H0: the variance in outdegree between the two genders is equal
outdegree_gender$gender <- as.factor(outdegree_gender$gender)

leveneTest(outdegree ~ gender,
           data = outdegree_gender)
```

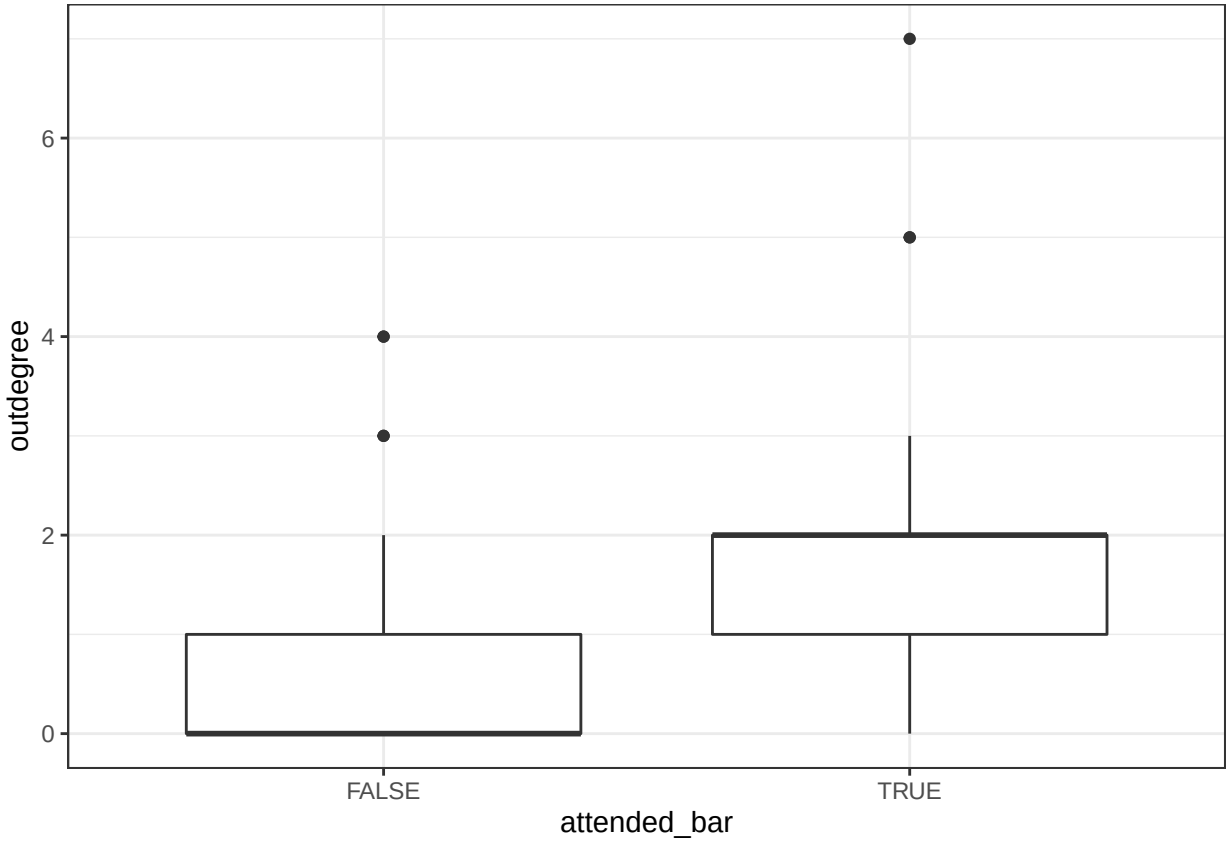
```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.2092 0.6486
##       84
```

```r
# p-value > 0.05, so we fail to reject H0, conclude the variances are equal

# Student's t-test comparing the mean outdegree of men and women
# H0: mu_x - mu_y = 0
t.test(x = outdegree_m,
       y = outdegree_f,
       var.equal = T, alternative = "two.sided")
```

```
##
##  Two Sample t-test
##
## data:  outdegree_m and outdegree_f
## t = 0.6411, df = 84, p-value = 0.5232
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.3910458  0.7631389
## sample estimates:
## mean of x mean of y
## 0.9767442 0.7906977
```

```
# p-value = 0.5232 > 0.05, we fail to reject H0
```

```
# boxplot comparing distribution of outdegree between bar (not)-attended
ggplot(gonnet_df2 %>% filter(!(nodes %in% c("b", "x", "x2"))),
       aes(x = attended_bar, outdegree)) +
  geom_boxplot() + theme_bw()
```



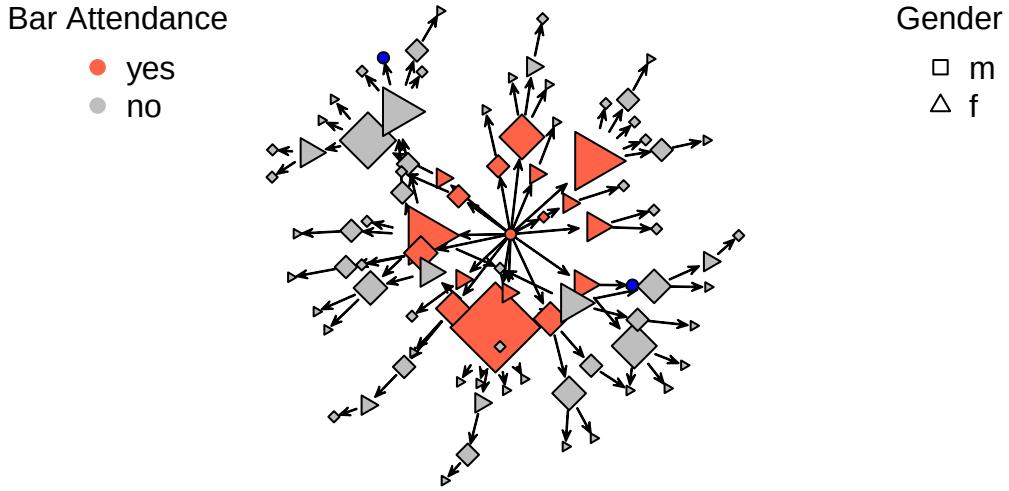

```
# average outdegree among bar attendees
outdegree_bar <- gonnet_df2 %>%
  select(nodes, attended_bar, outdegree) %>%
  filter(!(nodes %in% c("b", "x2", "x"))) %>%
  filter(attended_bar == TRUE)

# average outdegree among non-bar attendees
outdegree_nobar <- gonnet_df2 %>%
  select(nodes, attended_bar, outdegree) %>%
  filter(!(nodes %in% c("b", "x2", "x"))) %>%
  filter(attended_bar == FALSE)

# sample sizes are unequal, so we cannot assume equal variance
# try two-sided Welch's t.test
# H0: mu_x - mu_y = 0
t.test(x = outdegree_bar$outdegree,
       y = outdegree_nobar$outdegree,
       var.equal = FALSE, alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  outdegree_bar$outdegree and outdegree_nobar$outdegree
## t = 3.4857, df = 18.197, p-value = 0.002605
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.6408486 2.5816578
## sample estimates:
## mean of x mean of y
## 2.1764706 0.5652174
```

```
# p-value < 0.05, reject H0
# conclude the difference in mean outdegree is not equal to zero
```

```
# simulating similar networks and determining whether than outdegree is statistically significant
mc_sim(gonnet_sym, n = 1000, alpha = 0.05, "three-cycle")
mc_sim(gonnet_sym, n = 1000, alpha = 0.05, "mutuality")
mc_sim(gonnet_sym, n = 1000, alpha = 0.05, "transitivity")
```

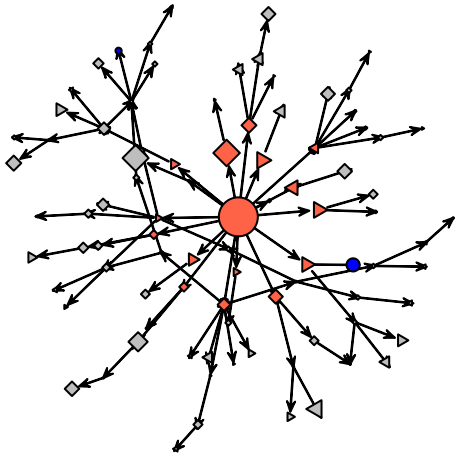
```
# normalizing the outdegree of bar because of course it has a high outdegree
outdegree_m <- cen$outdegree + 1
outdegree_m[1] <- 1

set.seed(10)

# plotting based on outdegree
gplot(gonnet,
      vertex.col = gonnet_df$col,
      vertex.sides = gonnet_df$gender_lty,
      vertex.cex = outdegree_m,
      coord = org_coord)
# legend for gender
legend("topleft",
       legend = c("yes", "no"),
       col = c("tomato1", "grey"),
       fill = F, border = "white", pch = 19,
       title = "Bar Attendance", bty = "n")
legend("topright",
       legend = c("m", "f"),
       col = c("black"),
       fill = F, border = "white", pch = c(0, 2),
       title = "Gender", bty = "n")
title("Gonorrhea network, sized by outdegree")
```



Principal Component Analysis

```
# extracting the centralities that were important based on the PCA
centrality_eigen <- centralities$`eigenvector centralities`
centrality_load <- centralities$`Load Centrality`
centrality_degree <- centralities$`Degree Centrality`
centrality_geodesic <- centralities$`Geodesic K-Path Centrality`
centrality_shortest <- centralities$`Shortest-Paths Betweenness Centrality`
centrality_info <- centralities$`Information Centrality`
```

```
set.seed(10)

centrality_degree[1] <- 1
gplot(gonnet, vertex.col = gonnet_df$col,
      vertex.sides = gonnet_df$gender_lty,
      vertex.cex = centrality_info %>% scale(),
      # displaylabels = T, label.cex = 0.6, label.pos = 2,
      boxed.labels = F, pad = 2,
      coord = org_coord)
```



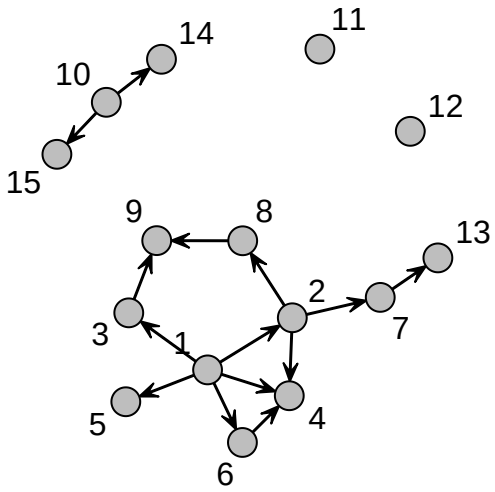
```
set.seed(10)
par(mfrow = c(1,1))
clust <- hclust(dist(gonnet), method = "complete")
# clust <- equiv.clust(gonnet)

plot(clust)
```


```
bm_complete <- blockmodel(gonnet,
                          ec = clust, k = 15)

bmR_complete <- reorder_blockmodel(bm_complete)
# overall network density, which we will use for alpha
alpha <- bmR_complete$block.model %>% mean(na.rm = T)
paste0("The mean overall network density is ", round(alpha, 3), ".")
```

```
## [1] "The mean overall network density is 0.056."
```

```
density_matrix <- bmR_complete$block.model
# plotting blockmodel
gplot(density_matrix > alpha,
      diag = T,
      vertex.cex = 1.5,
      label = unique(bmR_complete$block.membership),
      # boxed.labels=F,
      vertex.col="grey",
      pad = 1.25)
title("Block sociogram")
```

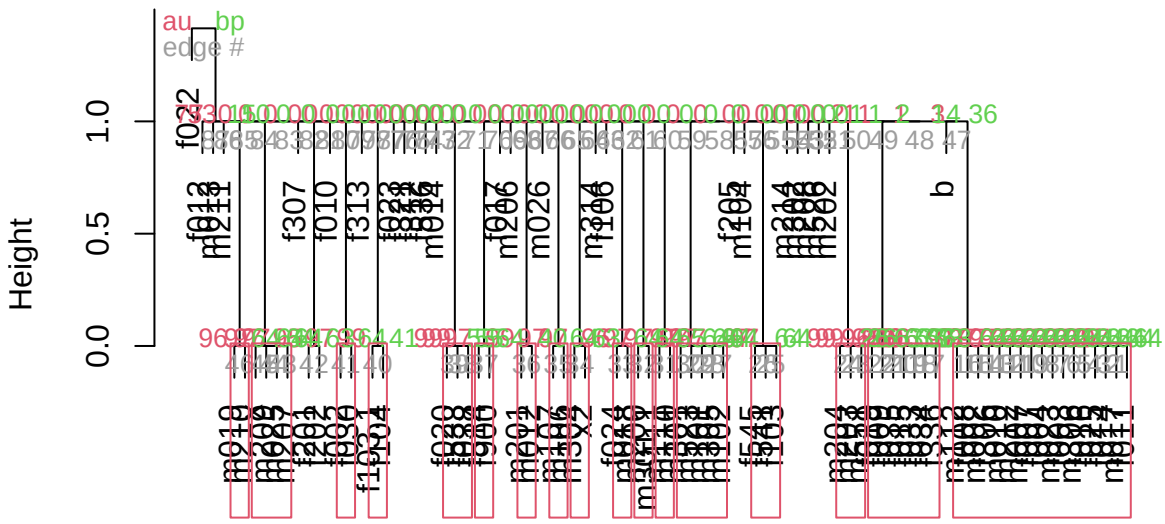
Block sociogram




```
fit <-
  pvclust(gonnet,
  method.hclust = "single",
  method.dist = "euclidean",
  iseed = 10, # to get same results
  parallel = T, # to use all but one CPU thread
  nboot = 1000)
```

```
## Creating a temporary cluster...done:
## socket cluster with 15 nodes on host 'localhost'
## Multiscale bootstrap... Done.
```

```
par(mfrow = c(1,1))
plot(fit)
pvrect(fit, alpha = 0.95)
```



Distance: euclidean
Cluster method: single

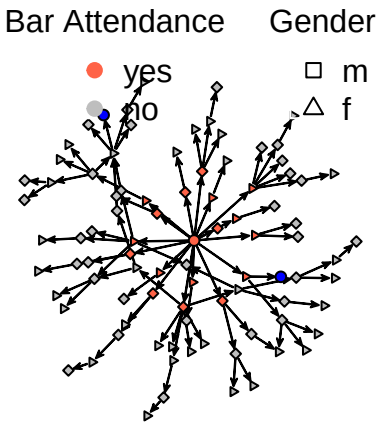
```
set.seed(10)
# if pchisq > 0.05, cluster is significant
pp <- pvpick(fit)
# number of significant clusters
ideal_k <- pp$clusters %>% length()


# blockmodel using k = 58 and cutting trees at 0.5
bm_complete <- blockmodel(gonnet,
                          ec = fit$hclust,
                          k = ideal_k+9, mode = "digraph")

bmR_complete <- reorder_blockmodel(bm_complete)
# overall network density, which we will use for alpha
alpha <- bmR_complete$block.model %>% mean(na.rm = T)
paste0("The mean overall network density is ", round(alpha, 3), ".")
```

```
## [1] "The mean overall network density is 0.013."
```

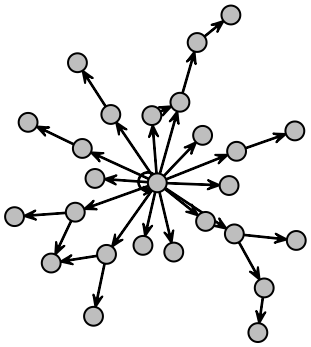
```
density_matrix <- bmR_complete$block.model

par(mfrow = c(1,2))
# original plot, again
gplot(gonnet, vertex.col = gonnet_df$col,
      vertex.sides = gonnet_df$gender_lty,
```

```
      # displaylabels = T, label.cex = 0.6, label.pos = 2,
      boxed.labels = F, pad = 2,
      vertex.cex = 1.25)
# legend for gender
legend("topleft",
       legend = c("yes", "no"),
       col = c("tomato1", "grey"),
       fill = F, border = "white", pch = 19,
       title = "Bar Attendance", bty = "n")
legend("topright",
       legend = c("m", "f"),
       col = c("black"),
       fill = F, border = "white", pch = c(0, 2),
       title = "Gender", bty = "n")
# legend for bar attendance
title("Network of gonorrhea transfer")

# plotting blockmodel
gplot(density_matrix > alpha,
      diag = T,
      vertex.cex = 1.5,
      # label = unique(bmR_complete$block.membership),
      # boxed.labels=F,
      vertex.col="grey",
      pad = 1.25,
      # coord = org_coord
      )
title("Block sociogram")
```

Network of gonorrhea transfer

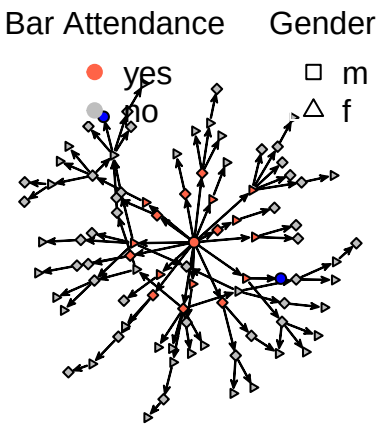


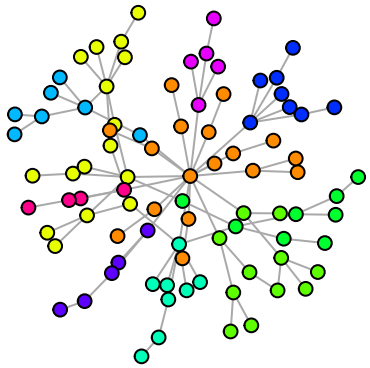
Block sociogram



```
set.seed(10)
par(mfrow = c(1,2))
# original plot, again
org_coord <- gplot(gonnet, vertex.col = gonnet_df$col,
      vertex.sides = gonnet_df$gender_lty,
      # displaylabels = T, label.cex = 0.6, label.pos = 2,
      boxed.labels = F, pad = 2,
      vertex.cex = 1.25)
# legend for gender
legend("topleft",
       legend = c("yes", "no"),
       col = c("tomato1", "grey"),
       fill = F, border = "white", pch = 19,
       title = "Bar Attendance", bty = "n")
legend("topright",
       legend = c("m", "f"),
       col = c("black"),
       fill = F, border = "white", pch = c(0, 2),
       title = "Gender", bty = "n")

# adding in shape arguments for igraph to correspond to original sna gplot
# View(gonnet_df)


# fast and greedy community detection
plot_fastgreedy_cd(gonnet, layout = org_coord,
```

```
                legend = F, vertex.size = 8)
```





```
# labeling only the original bar patrons
```

```
par(mfrow = c(1,1))
set.seed(10)
edgelist <- as.edgelist(gonnet, n = dim(gonnet)[1])
plot_kcores(edgelist, sym = F, mode = "digraph",
            coord = org_coord,
            cmode = "outdegree")
```