

```
#####GPS Network Analysis - February 2021
```

```
#Install packages
```

```
install.packages("missForest")  
install.packages("foreign")  
install.packages("IsingFit")  
install.packages("qgraph")  
install.packages("igraph")  
install.packages("bootnet")  
install.packages("NetworkComparisonTest")  
install.packages("Matrix")
```

```
#####
```

```
#Load packages
```

```
library(missForest)  
library(foreign)  
library(IsingFit)  
library(qgraph)  
library(igraph)  
library(bootnet)  
library(NetworkComparisonTest)  
library(Matrix)
```

```
#####
```

```
#Imputations for missing data
```

```
# Select file
```

```
file.choose()  
Data <- read.spss("FILE LOCATION",use.value.labels = F,to.data.frame = T)
```

```
# Check code for missing values:
```

```
Data$How_long_ago_CAT[Data$How_long_ago_CAT == '999' ]<- NA
```

```
# Check class of variables:
```

```
Data$PrivateorWork <- as.factor(Data$PrivateorWork)  
Data$SingleMultiple <- as.factor(Data$SingleMultiple)
```

```
class(Data$SingleMultiple)  
class(Data2$PrivateorWork)
```

```
# Ommit ID number for Missforest algorithm, add this in later
```

```
Data2 = subset(Data, select = -c(X._id) )
```

```
# MISFOREST:
```

```

Result <- missForest(Data2)
NewData <- Result$ximp

# Add ID number
NewData$X._id <- Data$X._id

#Prep data; create two groups: covid-related traumas (cov) and other traumas (oth).

GPS <- read.spss("FILE LOCATION", use.value.labels = TRUE,to.data.frame = TRUE)

vars <-
c("GPS1","GPS2","GPS3","GPS4","GPS5","GPS6","GPS7","GPS8","GPS9","GPS10","GPS11","GPS12","GPS
14","GPS15","GPS16","GPS18")

GPSdatsub <- NewData[vars]

covdat <- subset(NewData, CoronaCombined == "Corona related")
othdat <- subset(NewData, CoronaCombined == "Non-corona")

covdatsub <- covdat[vars]
othdatsub <- othdat[vars]

#Change variable names and create abbreviated symptom names
Names <- scan("~GPSitemNames.txt", what = "character")
FullNames <- scan("~GPSitemNamesFull.txt", what = "character")

#####
#Network estimation
#IsingFit (van Borkulo & Epskamp, 2015)

covnet <- IsingFit(covdatsub, family='binomial', AND = TRUE, gamma = 0.5, plot = FALSE, progressbar =
TRUE)
#The weight adjacency matrix
covnet$weiadj

othnet <- IsingFit(othdatsub, family='binomial', AND = TRUE, gamma = 0.5, plot = FALSE, progressbar =
TRUE)
#The weight adjacency matrix
othnet$weiadj

#compute average layout
L <- averageLayout(covplot, othplot)
layout(t(1:2))

```

```

#Estimated networks of both groups.
covplot = qgraph(covnet$weiadj, weighted = T, directed = F, title = "Covid", labels = Names, color =
"gold", label.scale.equal = TRUE, label.cex = 2, layout= L, legend= FALSE, theme = "colorblind")

othplot = qgraph(othnet$weiadj, weighted = T, directed = F, title = "Other", labels = Names, color =
"darkseagreen2", label.scale.equal = TRUE, label.cex = 2, layout= L, legend= FALSE, theme = "colorblind")

#####
#Community detection and plotting
#walktrap algorithm (Pons & Latapy, 2005) using igraph package(Amestoy et al., 2015).

covigraph<- graph_from_adjacency_matrix(abs(covnet$weiadj), "undirected", weighted = TRUE,
add.colnames = FALSE)
covcom <- cluster_walktrap(covigraph)
communities(covcom)
covcomplot <- qgraph(covnet$weiadj, layout = L, cut = .8, groups = communities(covcom), legend =
FALSE, theme = 'colorblind')

othigraph<- graph_from_adjacency_matrix(abs(othnet$weiadj), "undirected", weighted = TRUE,
add.colnames = FALSE)
othcom <- cluster_walktrap(othigraph)
communities(othcom)

#compute average layout
L <- averageLayout(covcomplot, othcomplot)
layout(t(1:2))
#Figure 1. Community structure of networks detected by the Walktrap algorithm.
covcomplot <- qgraph(covnet$weiadj, cut = .8, groups = communities(covcom),
layout = L, vsize = 8, labels = Names, label.scale.equal = TRUE,
label.scale = TRUE, label.cex = 1.4, title = "COVID19",
title.cex = .5, legend = FALSE, theme = 'colorblind')
othcomplot <- qgraph(othnet$weiadj, cut = .8, groups = communities(othcom),
layout = L, vsize = 8, labels = Names, label.scale.equal = TRUE,
label.scale = TRUE, label.cex = 1.4, title = "Other",
title.cex = .5, legend = FALSE, theme = 'colorblind')

tiff(filename="C:/Users/RWilliamson/Desktop/Figure1.tiff", width=1440,height=720, res=300)
layout(t(1:2))
qgraph(covnet$weiadj, cut = .8, groups = communities(covcom),
layout = L, vsize = 8, labels = Names, label.scale.equal = TRUE,
label.scale = TRUE, label.cex = 1.4, title = "COVID19",
title.cex = .5, legend = FALSE, theme = 'colorblind')
qgraph(othnet$weiadj, cut = .8, groups = communities(othcom),
layout = L, vsize = 8, labels = Names, label.scale.equal = TRUE,

```

```
label.scale = TRUE, label.cex = 1.4, title = "Other",
title.cex = .5, legend = FALSE, theme = 'colorblind')
dev.off()
```

```
#####
```

```
#Connectivity
```

```
covSPL <- centrality(covplot)$ShortestPathLengths
covSPL <- covSPL[upper.tri(covSPL)]
covASPL <- mean(covSPL) #3.17
```

```
othSPL <- centrality(othplot)$ShortestPathLengths
othSPL <- othSPL[upper.tri(othSPL)]
othASPL <- mean(othSPL) #2.99
```

```
#####
```

```
#Edge stability
```

```
set.seed(3)
covESta <- bootnet(covdatsub, 1000, default = 'IsingFit', type = "nonparametric")
covEStaplot <- plot(covESta, plot = 'interval', order = "sample", labels = FALSE)
```

```
set.seed(4)
othESta <- bootnet(othdatsub, 1000, default = 'IsingFit', type = "nonparametric")
othEStaplot <- plot(othESta, plot = 'interval', order = 'sample', labels= FALSE)
```

```
#Supplimentary Figure 1
```

```
#Plot of the stability of the edge estimates of each network. Points indicate the mean edge weights of
the bootstrapped samples and lines indicate the width of the 95% confidence interval.
```

```
plot(covESta, plot = 'interval', order = "sample", labels = FALSE)
plot(othESta, plot = 'interval', order = 'sample', labels= FALSE)
```

```
#Supplimentary Figure 2a/b
```

```
#Matrix of comparing all edges and significant testing
```

```
plot(covESta, "edge", "difference", onlyNonZero = TRUE, order = "sample")
plot(othESta, "edge", "difference", onlyNonZero = TRUE, order = "sample")
```

```
#####
```

```
#Centrality Stability
```

```
set.seed(1)
covCenSta <- bootnet(covdatsub, nBoots = 1000, default = "IsingFit", statistics = c("strength",
"betweenness", "closeness"), type = "case")
covCentStaplot <- plot(covCenSta, c("strength", "betweenness", "closeness"), subsetRange = c(100,25))
```

```
corStability(covCenSta) #strength= 0.517 between= .05 close= .361
```

```
set.seed(2)
```

```
othCenSta <- bootnet(othdatsub, nBoots = 1000, default = "IsingFit", statistics = c("strength",  
"betweenness", "closeness"), type = "case")
```

```
othCentStaplot <- plot(othCenSta, c("strength", "betweenness", "closeness"), subsetRange = c(100,25))
```

```
corStability(othCenSta) #strength= 0.75 between= .439 close= .75
```

```
#Supplementary Figure 3
```

```
#Stability plot of the centrality estimates of each network. The lines represent the mean correlations  
between the given sampled percentage and the complete sample and the shades represent the area  
between the 2.5% and 97.5% percentile of the sampled estimates.
```

```
plot(covCenSta, c("strength", "betweenness", "closeness"), subsetRange = c(100,25))
```

```
plot(othCenSta, c("strength", "betweenness", "closeness"), subsetRange = c(100,25))
```

```
#####
```

```
#Centrality
```

```
centRescov <- centrality(covplot)
```

```
centRescov$OutDegree
```

```
centRescov$Closeness
```

```
centRescov$Betweenness
```

```
covcenttab <- centralityTable(covplot, standardized = FALSE)
```

```
centResoth <- centrality(othplot)
```

```
centResoth$OutDegree
```

```
centResoth$Closeness
```

```
centResoth$Betweenness
```

```
othcenttab <- centralityTable(othplot, standardized = FALSE)
```

```
#Figure 2: Centrality indices of both networks.
```

```
centRescovoth = centralityPlot(list(COVID19 = covplot, Other = othplot),  
labels = Names, orderBy = "Strength", decreasing = FALSE,  
include = c("Closeness", "Strength"), theme_bw = TRUE, print = TRUE)
```

```
tiff(filename="~Fig2.tiff", width= 2404, height= 1600, units="px", res=300)
```

```
centralityPlot(list(COVID19 = covplot, Other = othplot),  
labels = Names, theme_bw = TRUE, orderBy = "Strength", decreasing = FALSE,  
include = c("Closeness", "Strength")
```

```
)
```

```
dev.off()
```

```
#####
```

```
#Network Comparison Test
```

```
set.seed(111)
```

```
NCTcovoth <- NCT(covdatsub, othdatsub, it=1000, binary.data= TRUE, paired = FALSE, test.edges=TRUE,  
edges="all", test.centralty=TRUE, centrality = c("strength"))
```

```
NCTcovoth$glstrinv.real #6.12
```

```
NCTcovoth$glstrinv.sep
```

```
NCTcovoth$glstrinv.pval #0.000
```

```
NCTcovoth$nwinv.real #0.478
```

```
NCTcovoth$nwinv.pval #0.253
```

```
NCTcovoth$einv.pvals #p values for each edge 2-5, 3-8, 3-10, 2-14, 14-15, 14-16
```

```
sum(NCTcovoth$einv.pvals$"p-value" < 0.05) #6
```

```
layout(t(1:2))
```

```
plot(NCTcovoth, what="strength")
```

```
plot(NCTcovoth, what="network")
```

```
inputNCTgraph <- covnet$weiadj - othnet$weiadj #creates input for edgediffplot, which is based on the  
differences in edge weights between the covid and other networks
```

```
inputNCTgraph[upper.tri(inputNCTgraph)][which(NCTcovoth$einv.pvals$"p-value" > .05)] <- 0 #sets the  
non-significant edge weight differences in the upper triangle of the input matrix to 0
```

```
inputNCTgraph <- forceSymmetric(inputNCTgraph) #sets the lower triangle of the matrix to be  
symmetric with the upper triangle
```

```
#Figure 3. Edges that differ significantly between the Covid and the Other network. Values indicate the  
difference between the edges in the Covid network and the Other network.
```

```
layout(1:1)
```

```
edgediffplot <- qgraph(inputNCTgraph, layout = L,  
labels = Names, label.scale.equal = TRUE,  
label.fill.horizontal = 1, label.cex = 2, edge.labels = TRUE,  
vsize = 6, esize = 10, label.font = 2, edge.label.position = .5,  
edge.label.cex = 1, theme = "colorblind", fade = FALSE,  
legend = FALSE)
```

```
tiff(filename=~"Fig3.tiff", width=1440,height=720, res=300)
```

```
edgediffplothr <- qgraph(inputNCTgraph, groups = communities(othcom),  
labels = Names, label.scale.equal = TRUE,  
label.fill.horizontal = 1, label.cex = 1.2, edge.labels = TRUE,  
vsize = 6, esize = 10, label.font = 2, edge.label.position = .5,  
edge.label.cex = 1, theme = "colorblind", fade = FALSE,  
legend = FALSE)
```

```
dev.off()
```

```
### NCT - bootstrapped samples of equal groups
# number of bootstrap samples
grpdraws = 10
#number of draws for each bootstrap sample - size of smaller group
grpN = 1699

#draw bootstrap samples and estimate networks
set.seed(555)
NCTboot = list()
othdatboot = list()
for (r in 1:grpdraws) {
  s1 = sample(1:nrow(othdatsub), grpN, replace = T)
  othdatboot[[r]] = othdatsub[s1,]
  NCTboot[[r]] = NCT(covdatsub, othdatboot[[r]], it=1000, binary.data= TRUE, paired = FALSE,
test.edges=TRUE, edges="all", test.centralitv=TRUE)
}
```