#COVID-19 Comparisonstudy 2020

#########################

#Install and load packages (uncomment if you have not installed them already)

#install.packages("haven")

library(haven)

#install.packages("mgm")

library(mgm)

#install.packages("Matrix")

library(Matrix)

#install.packages("qgraph")

library(qgraph)

#install.packages("igraph")

library(igraph)

#install.packages("NetworkComparisonTest")

library(NetworkComparisonTest)

#install.packages("plyr")

library(plyr)

#install.packages("dplyr")

library(dplyr)

#install.packages("bootnet")

library(bootnet)

#install.packages("openxlsx")

library(openxlsx)

#install.packages("matrixcalc")

library(matrixcalc)

#########################

#Setworkingdirectory

setwd("/.../COVID-19 Comparison study")

########################

#Create datasets

#Load dataset Final\_Merged dataset COVID-19 comparison study.sav

FullDataset <- read\_spss("/.../COVID-19 Comparison study/Final\_Merged dataset COVID-19 comparison study.sav")

View(FullDataset)

##Create function Community stability

communityStability <- function(data, type, level, iterations)

{

communityMemberships <- list()

for (i in 1:iterations)

{

fitGraph <- mgm(data, type, level, k = 2, verbatim = TRUE)

iGraph<- graph\_from\_adjacency\_matrix(abs

(fitGraph$pairwise$wadj), "undirected", weighted =

TRUE)

communityMemberships[[i]] <- cluster\_walktrap(iGraph)$membership

}

communityOverlap <- matrix(NA, length(data), length(data))

for(j in 1:length(data))

{

overlapPerNode <- matrix(NA, iterations, ncol(communityOverlap))

for(k in 1: iterations)

{

overlapPerNode[k,] <- as.numeric(communityMemberships[[k]][j] == communityMemberships[[k]])

}

communityOverlap[j,] <- apply(overlapPerNode, 2, mean)

}

diag(communityOverlap) <- 0

return(list(communityMemberships = communityMemberships, communityOverlap = communityOverlap))

}

######

#Create Partial dataset with relevant variables including factors to devide (contains missings)

PartialDataset <- FullDataset %>% select(

Contr\_vaccin ,

Demo\_age ,

Demo\_gender,

Demo\_health\_gen ,

Demo\_health\_physic ,

Demo\_health\_mental ,

Demo\_smoke,

Demo\_illnes ,

Sample,

Perceived\_Infection ,

InfectionNearby ,

RiskPerception ,

HealthRisk ,

ConsequencesSociety ,

ConsequencesEconomy ,

AffectNegative ,

AffectPositive ,

WorryPersonal ,

WorrySociety ,

TrustAuthorities ,

Humanitarianism ,

NeedForChaos,

MeasuresSupport ,

MeasuresEfficacy ,

NormSociety ,

NormFamilyFriends ,

ControlInfection ,

CompliancePrevent ,

ComplianceRepress)

#Recode binairy from 1 and 2 to 0 and 1 for binarySign in mgm

#Demo\_gender (from 1 = male & 2 = female) to 0 = male, 1 = female

PartialDataset$Demo\_gender[PartialDataset$Demo\_gender == 1] <- 0

PartialDataset$Demo\_gender[PartialDataset$Demo\_gender == 2] <- 1

#Demo\_smoke & Demo\_illness (from 1 = yes and 2 = no) to 1 = yes, 0 = no

PartialDataset$Demo\_smoke[PartialDataset$Demo\_smoke == 2] <- 0

PartialDataset$Demo\_illnes[PartialDataset$Demo\_illnes == 2] <- 0

#dataset specific for CAN - same as full but without Perceived\_Infection & InfectionNearby

CANDataset <- PartialDataset %>% select(-Perceived\_Infection)

CANDataset <- CANDataset %>% select(-InfectionNearby)

CANDataset <- na.omit(CANDataset) ##delete missings casewise

View(CANDataset)

#create object data Total

DataCovidTotal<-CANDataset %>% select(-Sample)

View(DataCovidTotal)

#create object data UK

DataCovidUK<- subset(CANDataset, Sample ==1)

DataCovidUK<- DataCovidUK %>% select(-Sample)

View(DataCovidUK)

#create object data NL

DataCovidNL<- subset(CANDataset, Sample ==2)

DataCovidNL<- DataCovidNL %>% select(-Sample)

View(DataCovidNL)

#Create CAN Total

summary(DataCovidTotal)

#create type

TypeCovidTotal<-c(

"g","g","c","g","g",

"g","c","c","g","g",

"g","g","g","g","g",

"g","g","g","g","g",

"g","g","g","g","g","g")

#create level

LevelCovidTotal <- c(

1,1,2,1,1,

1,2,2,1,1,

1,1,1,1,1,

1,1,1,1,1,

1,1,1,1,1,1)

#create nodeNames for legend

SHORTColnamesCovidTotal<-c(

"Vac",

"Age",

"Gen^",

"H\_Gen" ,

"H\_Phys" ,

"H\_Ment" ,

"Smo^",

"Ill^",

"Risk\_P",

"H\_Risk",

"Con\_Soc",

"Con\_Eco",

"Aff\_Neg" ,

"Aff\_Pos" ,

"Wor\_Pers",

"Wor\_Soc",

"Trust\_Aut",

"Huma" ,

"Chaos" ,

"Meas\_Sup" ,

"Meas\_Eff" ,

"Norm\_Soc" ,

"Norm\_FF" ,

"Contr\_Inf",

"Prev\_Beh",

"Repr\_Beh")

#create nodeNames for legend

ColnamesCovidTotal<-c(

"Vaccine Intention",

"Age",

"Gender",

"Health General" ,

"Health Physical" ,

"Health Mental" ,

"Smoking",

"Illness",

"Risk Perception",

"Health Risk",

"Consequences Society",

"Consequences Economy",

"Affect Negative" ,

"Affect Positive" ,

"Worry Personal",

"Worry Society ",

"Trust Authorities" ,

"Humanitarianism" ,

"Need For Chaos" ,

"Measures Support" ,

"Measures Efficacy" ,

"Norm Society" ,

"Norm Family Friends" ,

"Control Infection",

"Preventive Behaviors",

"Repressive Behaviors")

#Estimating Total network mgm (lineair and binair combined)

set.seed(1)

FitCANCovidTotal <- mgm(DataCovidTotal, TypeCovidTotal, LevelCovidTotal, k = 2, binarySign = TRUE)

######Community detection and stability

CommunityStabTotal<-communityStability(DataCovidTotal, TypeCovidTotal, LevelCovidTotal, it=1000)

### Visualize community stability

pdf('Community Stability\_Totalsample.pdf',paper = "USr", height = 9, width = 12)

qgraph(CommunityStabTotal$communityOverlap, layout = "spring", theme = "colorblind",

labels = SHORTColnamesCovidTotal, nodeNames = ColnamesCovidTotal,vsize=6.0,

title = "Community Stability Total sample", edge.labels=FALSE, legend = TRUE, legend.cex = 0.3)

dev.off()

##Detection community stability

inputCommDetection\_total<-CommunityStabTotal$communityOverlap

inputCommDetection\_total[which(inputCommDetection\_total <= .90)] <- 0

pdf('Community Stability\_Totalsample\_detect.pdf',paper = "USr", height = 9, width = 12)

qgraph(inputCommDetection\_total, layout = "spring", theme = "colorblind",

labels = SHORTColnamesCovidTotal, nodeNames = ColnamesCovidTotal,vsize=6.0,

title = "Community Stability Total sample", edge.labels=FALSE, legend = TRUE, legend.cex = 0.3)

dev.off()

# group variables based on outcome detection community stability so we can assign colors

Totalgroup\_comm <- list(

" "=c(3,11,12,13,15,16),

" "=c(4,8,9,10),

" "=c(5,6),

" "=c(7,18,19),

" "=c(17,22,23,25,26),

" "=c(20,21),

" "=c(1,2,14,24))

# define nice colors

Totalgroup\_cols <- c("#E8ED61","#FFC237","#E35959","#4B71B3","#8FC45A","#CC79A7","#FFFFFF")

#create graph Total

#enables theme colorblind because we don't need to specify edge.color

inputGraphCANCovidTotal <- FitCANCovidTotal$pairwise$wadj

signsGraphCANCovidTotal <- FitCANCovidTotal$pairwise$signs

signsGraphCANCovidTotal[which(is.na(signsGraphCANCovidTotal))] <- 1

inputGraphCANCovidTotal <- inputGraphCANCovidTotal\*signsGraphCANCovidTotal

GraphCANCovidTotal<-qgraph(inputGraphCANCovidTotal,

layout = "spring", theme = "colorblind",

labels = SHORTColnamesCovidTotal,nodeNames = ColnamesCovidTotal,minimum = 0,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups=Totalgroup\_comm, color= Totalgroup\_cols, legend = TRUE, legend.cex = 0.4, title = "Total")

#########################################################

#Create network UK

summary(DataCovidUK)

#create type

TypeCovidUK<-c(

"g","g","c","g","g",

"g","c","c","g","g",

"g","g","g","g","g",

"g","g","g","g","g",

"g","g","g","g","g","g")

#create level

LevelCovidUK <- c(

1,1,2,1,1,

1,2,2,1,1,

1,1,1,1,1,

1,1,1,1,1,

1,1,1,1,1,1)

#create nodeNames for legend

SHORTColnamesCovidUK<-c(

"Vac",

"Age",

"Gen^",

"H\_Gen" ,

"H\_Phys" ,

"H\_Ment" ,

"Smo^",

"Ill^",

"Risk\_P",

"H\_Risk",

"Con\_Soc",

"Con\_Eco",

"Aff\_Neg" ,

"Aff\_Pos" ,

"Wor\_Pers",

"Wor\_Soc",

"Trust\_Aut",

"Huma" ,

"Chaos" ,

"Meas\_Sup" ,

"Meas\_Eff" ,

"Norm\_Soc" ,

"Norm\_FF" ,

"Contr\_Inf",

"Prev\_Beh",

"Repr\_Beh")

#create nodeNames for legend

ColnamesCovidUK<-c(

"Vaccine Intention",

"Age",

"Gender",

"Health General" ,

"Health Physical" ,

"Health Mental" ,

"Smoking",

"Illness",

"Risk Perception",

"Health Risk",

"Consequences Society",

"Consequences Economy",

"Affect Negative" ,

"Affect Positive" ,

"Worry Personal",

"Worry Society ",

"Trust Authorities" ,

"Humanitarianism" ,

"Need For Chaos" ,

"Measures Support" ,

"Measures Efficacy" ,

"Norm Society" ,

"Norm Family Friends" ,

"Control Infection",

"Preventive Behaviors",

"Repressive Behaviors")

#Estimating UK network mgm (lineair and binair combined)

set.seed(1)

FitCANCovidUK <- mgm(DataCovidUK, TypeCovidUK, LevelCovidUK, k = 2, binarySign = TRUE)

###### Community detection and stability UK

CommunityStabUK<-communityStability(DataCovidUK, TypeCovidUK, LevelCovidUK, it=1000)

###

pdf('Community Stability\_UK.pdf',paper = "USr", height = 9, width = 12)

qgraph(CommunityStabUK$communityOverlap, layout = "spring", theme = "colorblind",

labels = SHORTColnamesCovidUK, nodeNames = ColnamesCovidUK,vsize=6.0,

title = "Community Stability UK", edge.labels=FALSE, legend = TRUE, legend.cex = 0.3)

dev.off()

##Detection community stability UK

inputCommDetection\_UK<-CommunityStabUK$communityOverlap

inputCommDetection\_UK[which(inputCommDetection\_UK <= .90)] <- 0

pdf('Community Stability\_UK\_detect.pdf',paper = "USr", height = 9, width = 12)

qgraph(inputCommDetection\_UK, layout = "spring", theme = "colorblind",

labels = SHORTColnamesCovidUK, nodeNames = ColnamesCovidUK,vsize=6.0,

title = "Community Stability UK", edge.labels=FALSE, legend = TRUE, legend.cex = 0.3)

dev.off()

# community UK

# group variables based on outcome detection community stability so we can assign colors

UKgroup\_comm <- list( " "=c(1, 7, 17, 18, 19, 20, 21, 22),

" " = c(2, 14),

" "=c(4, 8, 9, 10),

" " = c(5, 6),

" "=c(11, 12, 13, 15, 16),

" " = c(23, 25, 26),

" " = c(3, 24))

# define colors

UKgroup\_cols <- c("#4B71B3","#CC79A7", "#FFC237","#E35959","#E8ED61","#8FC45A","#FFFFFF")

#create graph UK

#enables theme colorblind because we don't need to specify edge.color

inputGraphCANCovidUK <- FitCANCovidUK$pairwise$wadj

signsGraphCANCovidUK <- FitCANCovidUK$pairwise$signs

signsGraphCANCovidUK[which(is.na(signsGraphCANCovidUK))] <- 1

inputGraphCANCovidUK <- inputGraphCANCovidUK\*signsGraphCANCovidUK

GraphCANCovidUK<-qgraph(inputGraphCANCovidUK,

layout = "spring", theme = "colorblind",

labels = SHORTColnamesCovidUK,nodeNames = ColnamesCovidUK, minimum = 0,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

color= UKgroup\_cols, groups= UKgroup\_comm, legend = TRUE,legend.cex = 0.4, title = "UK")

########################################

#Create network NL

summary(DataCovidNL)

#create type

TypeCovidNL<-c(

"g","g","c","g","g",

"g","c","c","g","g",

"g","g","g","g","g",

"g","g","g","g","g",

"g","g","g","g","g","g")

#create level

LevelCovidNL <- c(

1,1,2,1,1,

1,2,2,1,1,

1,1,1,1,1,

1,1,1,1,1,

1,1,1,1,1,1)

#create nodeNames for legend

SHORTColnamesCovidNL<-c(

"Vac",

"Age",

"Gen^",

"H\_Gen" ,

"H\_Phys" ,

"H\_Ment" ,

"Smo^",

"Ill^",

"Risk\_P",

"H\_Risk",

"Con\_Soc",

"Con\_Eco",

"Aff\_Neg" ,

"Aff\_Pos" ,

"Wor\_Pers",

"Wor\_Soc",

"Trust\_Aut",

"Huma" ,

"Chaos" ,

"Meas\_Sup" ,

"Meas\_Eff" ,

"Norm\_Soc" ,

"Norm\_FF" ,

"Contr\_Inf",

"Prev\_Beh",

"Repr\_Beh")

#create nodeNames for legend

ColnamesCovidNL<-c(

"Vaccine Intention",

"Age",

"Gender",

"Health General" ,

"Health Physical" ,

"Health Mental" ,

"Smoking",

"Illness",

"Risk Perception",

"Health Risk",

"Consequences Society",

"Consequences Economy",

"Affect Negative" ,

"Affect Positive" ,

"Worry Personal",

"Worry Society ",

"Trust Authorities" ,

"Humanitarianism" ,

"Need For Chaos" ,

"Measures Support" ,

"Measures Efficacy" ,

"Norm Society" ,

"Norm Family Friends" ,

"Control Infection",

"Preventive Behaviors",

"Repressive Behaviors")

#Estimating NL network mgm (lineair and binair combined)

set.seed(1)

FitCANCovidNL <- mgm(DataCovidNL, TypeCovidNL, LevelCovidNL, k = 2, binarySign = TRUE)

##############################

##Community detection and stability NL

CommunityStabNL<-communityStability(DataCovidNL, TypeCovidNL, LevelCovidNL, it=1000)

###

pdf('Community Stability\_NL.pdf',paper = "USr", height = 9, width = 12)

qgraph(CommunityStabNL$communityOverlap, layout = "spring", theme = "colorblind",

labels = SHORTColnamesCovidNL, nodeNames = ColnamesCovidNL,

title = "Community Stability NL", edge.labels=FALSE, legend = TRUE, legend.cex = 0.3)

dev.off()

##Detection community stability NL

inputCommDetection\_NL<-CommunityStabNL$communityOverlap

inputCommDetection\_NL[which(inputCommDetection\_NL <= .90)] <- 0

pdf('Community Stability\_NL\_detect.pdf',paper = "USr", height = 9, width = 12)

qgraph(inputCommDetection\_NL, layout = "spring", theme = "colorblind",

labels = SHORTColnamesCovidNL, nodeNames = ColnamesCovidNL,

title = "Community Stability NL", edge.labels=FALSE, legend = TRUE, legend.cex = 0.3)

dev.off()

# community NL

# group variables based on outcome detection community stability so we can assign colors

NLgroup\_comm <- list( " "=c(1, 17, 18, 20, 21, 22, 23, 25, 26),

" " = c(2, 3, 4, 8),

" "=c(9, 10, 24),

" " = c(5, 6),

" " = c(7, 19),

" "=c(11, 12, 13, 15, 16),

" " = c(14))

# define colors

NLgroup\_cols <- c("#8FC45A","#CC79A7","#FFC237","#E35959", "#4B71B3","#E8ED61","#FFFFFF")

#create graph NL

#enables theme colorblind because we don't need to specify edge.color

inputGraphCANCovidNL <- FitCANCovidNL$pairwise$wadj

signsGraphCANCovidNL <- FitCANCovidNL$pairwise$signs

signsGraphCANCovidNL[which(is.na(signsGraphCANCovidNL))] <- 1

inputGraphCANCovidNL <- inputGraphCANCovidNL\*signsGraphCANCovidNL

GraphCANCovidNL<-qgraph(inputGraphCANCovidNL,

layout = "spring", theme = "colorblind",

labels = SHORTColnamesCovidNL,nodeNames = ColnamesCovidNL, minimum = 0,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups = NLgroup\_comm, color= NLgroup\_cols, legend = TRUE, legend.cex = 0.4, title = "NL")

#####################################

#plot comparison UK NL

pdf('Figure 5 Compare networks UK NL.pdf', paper = "USr", height = 9, width = 12)

L<-averageLayout(GraphCANCovidTotal,GraphCANCovidUK, GraphCANCovidNL)

lmat <- matrix(1:2, 1, 2)

lo <- layout(lmat, width = c(1, 1.3))

GraphCANCovidUK<-qgraph(inputGraphCANCovidUK,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidUK,nodeNames = ColnamesCovidUK, minimum = 0,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups= UKgroup\_comm, color= UKgroup\_cols, legend = FALSE, title = "UK")

GraphCANCovidNL<-qgraph(inputGraphCANCovidNL,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidNL,nodeNames = ColnamesCovidNL, minimum = 0,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups = NLgroup\_comm, color= NLgroup\_cols, legend = TRUE, legend.cex = 0.25, title = "NL")

dev.off()

pdf('Figure 5 Compare networks UK NL\_min.05.pdf', paper = "USr", height = 9, width = 12)

lmat <- matrix(1:2, 1, 2)

lo <- layout(lmat, width = c(1, 1.3))

GraphCANCovidUK\_min.05<-qgraph(inputGraphCANCovidUK,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidUK,nodeNames = ColnamesCovidUK, minimum = 0.05,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups= UKgroup\_comm, color= UKgroup\_cols, legend = FALSE, title = "UK")

GraphCANCovidNL\_min.05<-qgraph(inputGraphCANCovidNL,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidNL,nodeNames = ColnamesCovidNL, minimum = 0.05,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups = NLgroup\_comm, color= NLgroup\_cols, legend = TRUE, legend.cex = 0.25, title = "NL")

dev.off()

pdf('Figure 5 Compare networks UK NL\_min.10.pdf', paper = "USr", height = 9, width = 12)

lmat <- matrix(1:2, 1, 2)

lo <- layout(lmat, width = c(1, 1.3))

GraphCANCovidUK\_min.10<-qgraph(inputGraphCANCovidUK,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidUK,nodeNames = ColnamesCovidUK, minimum = 0.10,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups= UKgroup\_comm, color= UKgroup\_cols, legend = FALSE, title = "UK")

GraphCANCovidNL\_min.10<-qgraph(inputGraphCANCovidNL,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidNL,nodeNames = ColnamesCovidNL, minimum = 0.10,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups = NLgroup\_comm, color= NLgroup\_cols, legend = TRUE, legend.cex = 0.25, title = "NL")

dev.off()

pdf('Figure 5 Compare networks UK NL\_no legend.pdf', paper = "USr", height = 9, width = 12)

lmat <- matrix(1:2, 1, 2)

lo2 <- layout(lmat, width = c(1, 1))

GraphCANCovidUK<-qgraph(inputGraphCANCovidUK,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidUK,nodeNames = ColnamesCovidUK, minimum = 0,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups= UKgroup\_comm, color= UKgroup\_cols, legend = FALSE, title = "UK")

GraphCANCovidNL<-qgraph(inputGraphCANCovidNL,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidNL,nodeNames = ColnamesCovidNL, minimum = 0,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups = NLgroup\_comm, color= NLgroup\_cols, legend = FALSE, legend.cex = 0.25, title = "NL")

dev.off()

pdf('Figure 5 Compare networks UK NL\_no legend\_min.05.pdf', paper = "USr", height = 9, width = 12)

lmat <- matrix(1:2, 1, 2)

lo2 <- layout(lmat, width = c(1, 1))

GraphCANCovidUK\_min.05<-qgraph(inputGraphCANCovidUK,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidUK,nodeNames = ColnamesCovidUK, minimum = 0.05,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups= UKgroup\_comm, color= UKgroup\_cols, legend = FALSE, title = "UK")

GraphCANCovidNL\_min.05<-qgraph(inputGraphCANCovidNL,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidNL,nodeNames = ColnamesCovidNL, minimum = 0.05,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups = NLgroup\_comm, color= NLgroup\_cols, legend = FALSE, legend.cex = 0.25, title = "NL")

dev.off()

pdf('Figure 5 Compare networks UK NL\_no legend\_min.10.pdf', paper = "USr", height = 9, width = 12)

lmat <- matrix(1:2, 1, 2)

lo2 <- layout(lmat, width = c(1, 1))

GraphCANCovidUK\_min.10<-qgraph(inputGraphCANCovidUK,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidUK,nodeNames = ColnamesCovidUK, minimum = 0.10,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups= UKgroup\_comm, color= UKgroup\_cols, legend = FALSE, title = "UK")

GraphCANCovidNL\_min.10<-qgraph(inputGraphCANCovidNL,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidNL,nodeNames = ColnamesCovidNL, minimum = 0.10,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups = NLgroup\_comm, color= NLgroup\_cols, legend = FALSE, legend.cex = 0.25, title = "NL")

dev.off()

#plot graphs seperate

layout(t(1:2))

GraphCANCovidUK<-qgraph(inputGraphCANCovidUK,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidUK,nodeNames = ColnamesCovidUK, minimum = 0,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups= UKgroup\_comm, color= UKgroup\_cols, legend = TRUE, legend.cex = 0.4, title = "UK", filetype = "pdf", filename = "NetworkCovid\_UK")

GraphCANCovidNL<-qgraph(inputGraphCANCovidNL,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidNL,nodeNames = ColnamesCovidNL, minimum = 0,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups = NLgroup\_comm, color= NLgroup\_cols, legend = TRUE, legend.cex = 0.4, title = "NL", filetype = "pdf", filename = "NetworkCovid\_NL")

pdf('Figure 3 Network COVID-19\_totalsample.pdf', paper = "USr", height = 17, width = 12)

GraphCANCovidTotal<-qgraph(inputGraphCANCovidTotal,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidTotal,nodeNames = ColnamesCovidTotal, minimum = 0.0,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups=Totalgroup\_comm, color= Totalgroup\_cols,legend = TRUE, legend.cex = 0.4, title = "Total sample")

dev.off()

#plot graphs seperate\_min.05

layout(t(1:2))

GraphCANCovidUK\_min.05<-qgraph(inputGraphCANCovidUK,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidUK,nodeNames = ColnamesCovidUK, minimum = 0.05,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups= UKgroup\_comm, color= UKgroup\_cols, legend = TRUE, legend.cex = 0.4, title = "UK", filetype = "pdf", filename = "NetworkCovid\_UK")

GraphCANCovidNL\_min.05<-qgraph(inputGraphCANCovidNL,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidNL,nodeNames = ColnamesCovidNL, minimum = 0.05,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups = NLgroup\_comm, color= NLgroup\_cols, legend = TRUE, legend.cex = 0.4, title = "NL", filetype = "pdf", filename = "NetworkCovid\_NL")

pdf('Figure 3 Network COVID-19\_totalsample\_min.05.pdf', paper = "USr", height = 17, width = 12)

GraphCANCovidTotal\_min.05<-qgraph(inputGraphCANCovidTotal,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidTotal,nodeNames = ColnamesCovidTotal, minimum = 0.05,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups=Totalgroup\_comm, color= Totalgroup\_cols,legend = TRUE, legend.cex = 0.4, title = "Total sample")

dev.off()

#plot graphs seperate\_min.10

layout(t(1:2))

GraphCANCovidUK\_min.10<-qgraph(inputGraphCANCovidUK,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidUK,nodeNames = ColnamesCovidUK, minimum = 0.10,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups= UKgroup\_comm, color= UKgroup\_cols, legend = TRUE, legend.cex = 0.4, title = "UK", filetype = "pdf", filename = "NetworkCovid\_UK")

GraphCANCovidNL\_min.10<-qgraph(inputGraphCANCovidNL,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidNL,nodeNames = ColnamesCovidNL, minimum = 0.10,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups = NLgroup\_comm, color= NLgroup\_cols, legend = TRUE, legend.cex = 0.4, title = "NL", filetype = "pdf", filename = "NetworkCovid\_NL")

pdf('Figure 3 Network COVID-19\_totalsample\_min.10.pdf', paper = "USr", height = 17, width = 12)

GraphCANCovidTotal\_min.10<-qgraph(inputGraphCANCovidTotal,

layout = L, theme = "colorblind",

labels = SHORTColnamesCovidTotal,nodeNames = ColnamesCovidTotal, minimum = 0.10,

cut = 0.10, maximum = 1, details = TRUE,vsize=6.0,

groups=Totalgroup\_comm, color= Totalgroup\_cols,legend = TRUE, legend.cex = 0.4, title = "Total sample")

dev.off()

##Matrix Edge Weights, write to Excel

EdgeWeightsExcel<- list("Total" = inputGraphCANCovidTotal,

"UK" = inputGraphCANCovidUK, "NL" = inputGraphCANCovidNL)

write.xlsx(EdgeWeightsExcel, "EdgeWeightsExcel.xlsx", colWidths = "auto", rowNames = TRUE)

##only upper triangle without edge weigths 0 for readability

EdgeWeight\_Total\_half<-upper.triangle(inputGraphCANCovidTotal)

EdgeWeight\_UK\_half<-upper.triangle(inputGraphCANCovidUK)

EdgeWeight\_NL\_half<-upper.triangle(inputGraphCANCovidNL)

EdgeWeight\_Total\_half[EdgeWeight\_Total\_half == 0] <- NA

EdgeWeight\_UK\_half[EdgeWeight\_UK\_half == 0] <- NA

EdgeWeight\_NL\_half[EdgeWeight\_NL\_half == 0] <- NA

EdgeWeightsExcel\_half<- list("Total" = EdgeWeight\_Total\_half,

"UK" = EdgeWeight\_UK\_half, "NL" = EdgeWeight\_NL\_half)

write.xlsx(EdgeWeightsExcel\_half, "EdgeWeightsExcel\_half.xlsx", colWidths = "auto", rowNames = TRUE)

##############################

#centrality measures raw data

#graph with complete labels for centrality plot

GraphCANCovidTotalCENT<-qgraph(inputGraphCANCovidTotal,

layout = L, theme = "colorblind",

labels = ColnamesCovidTotal, minimum = 0,

cut = 0.10, maximum = 1, details = TRUE,

groups=Totalgroup\_comm, color= Totalgroup\_cols,legend = FALSE, title = "Total sample")

#centrality Total

centCANCovidTotal <- centralityTable(GraphCANCovidTotalCENT,standardized = FALSE)

#CentralityPlot Total

pdf('Figure 4 Centrality plot Strength\_totalsample.pdf')

centralityPlot(GraphCANCovidTotalCENT, include = c("Strength"),scale = "raw")

dev.off()

#UK

#graph with complete labels for centrality plot

GraphCANCovidUKCENT<-qgraph(inputGraphCANCovidUK,

layout = L, theme = "colorblind",

labels = ColnamesCovidUK, minimum = 0,

cut = 0.10, maximum = 1, details = TRUE,

groups= UKgroup\_comm, color= UKgroup\_cols, legend = FALSE, title = "UK")

#centrality UK

centCANCovidUK <- centralityTable(GraphCANCovidUKCENT,standardized = FALSE)

#CentralityPlot UK

pdf('CentralityPlot Strength\_UK.pdf')

centralityPlot(GraphCANCovidUKCENT, include = c("Strength"),scale = "raw")

dev.off()

#NL

#graph with complete labels for centrality plot

GraphCANCovidNLCENT<-qgraph(inputGraphCANCovidNL,

layout = L, theme = "colorblind",

labels = ColnamesCovidNL, minimum = 0,

cut = 0.10, maximum = 1, details = TRUE,

groups = NLgroup\_comm, color= NLgroup\_cols, legend = FALSE, title = "NL")

#centrality NL

centCANCovidNL <- centralityTable(GraphCANCovidNLCENT,standardized = FALSE)

#CentralityPlot NL

pdf('CentralityPlot Strength\_NL.pdf')

centralityPlot(GraphCANCovidNLCENT, include = c("Strength"),scale = "raw")

dev.off()

#Plot centrality compare

pdf('Figure 7 Centrality plot comparison UK NL.pdf')

centralityPlot(list(UK=GraphCANCovidUKCENT,NL=GraphCANCovidNLCENT), include = c("Strength"),scale = "raw")

dev.off()

############ Optional with Z scores

#Centrality measures Z scores

#Plot UK Z

#pdf('CentralityPlot\_CANCovidUKZ.pdf')

#centralityPlot(GraphCANCovidUK, include = c("Strength"))

#dev.off

#Plot NL Z

#pdf('CentralityPlot\_CANCovidNZL.pdf')

#centralityPlot(GraphCANCovidNL, include = c("Strength"))

#dev.off()

#Plot centrality compare Z

#pdf('CentralityPlot\_CANCovidCompareZ.pdf')

#centralityPlot(list(UK=GraphCANCovidUK,NL=GraphCANCovidNL),include = c("Strength"))

#dev.off()

############################################

#Shortest Path Lenght

#UK

SPLCANCovidUK <- centrality(GraphCANCovidUK)$ShortestPathLengths

SPLCANCovidUK <- SPLCANCovidUK[upper.tri(SPLCANCovidUK)]

ASPLCANCovidUK <- mean(SPLCANCovidUK)

#NL

SPLCANCovidNL <- centrality(GraphCANCovidNL)$ShortestPathLengths

SPLCANCovidNL <- SPLCANCovidNL[upper.tri(SPLCANCovidNL)]

ASPLCANCovidNL <- mean(SPLCANCovidNL)

###########################################

#Network comparison test

set.seed(1)

names(DataCovidUK) <- names(DataCovidNL)

NCTCovidUKNL <- NCT(DataCovidUK, DataCovidNL, it = 1000, test.edges = TRUE, edges = "all")

# Difference in global strength p-value:

NCTCovidUKNL$glstrinv.real

NCTCovidUKNL$glstrinv.pval

# Maximum difference in edge weights:

NCTCovidUKNL$nwinv.real

NCTCovidUKNL$nwinv.pval

# Which edges significantly differ?

NCTCovidUKNL$einv.pvals[which(NCTCovidUKNL$einv.pvals[,3] < 0.05), ]

#input for NCTgraph

inputNCTgraph <- inputGraphCANCovidUK - inputGraphCANCovidNL

inputNCTgraph[upper.tri(inputNCTgraph)][which(NCTCovidUKNL$einv.pvals$`p-value` >= .05)] <- 0 #this sets the not-significant edge weight differences in the upper triangle of the input matrix to 0

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

pdf('Figure 6 NetworkComparisonTest UK NL.pdf',paper = "USr", height = 9, width = 12)

qgraph(inputNCTgraph, layout = GraphCANCovidTotal$layout, theme = "colorblind",

labels = SHORTColnamesCovidTotal, nodeNames = ColnamesCovidTotal,vsize=6.0,

title = "NCT", edge.labels=FALSE, legend = TRUE, legend.cex = 0.3)

dev.off()

pdf('Figure 6 NetworkComparisonTest UK NL\_labels.pdf',paper = "USr", height = 9, width = 12)

qgraph(inputNCTgraph, layout = GraphCANCovidTotal$layout, theme = "colorblind",

labels = SHORTColnamesCovidTotal, nodeNames = ColnamesCovidTotal,vsize=6.0,

title = "NCT", edge.labels=TRUE, legend = TRUE, legend.cex = 0.3)

dev.off()

DiffUKNL<- inputGraphCANCovidUK - inputGraphCANCovidNL

NCTExcel <- inputGraphCANCovidUK - inputGraphCANCovidNL

NCTExcel[upper.tri(NCTExcel)][which(NCTCovidUKNL$einv.pvals$`p-value` >= .05)] <- 0 #this sets the not-significant edge weight differences in the upper triangle of the input matrix to 0

NCTExcel<-upper.triangle(NCTExcel)

NCTExcel[NCTExcel == 0] <- NA

ExcelNCTTotal<- list("UK" = inputGraphCANCovidUK, "NL" = inputGraphCANCovidNL, "diff" = DiffUKNL, "NCT" = NCTExcel)

write.xlsx(ExcelNCTTotal,"NCT edge difference.xlsx", colWidths = "auto", rowNames = TRUE)

#######Stability

#Say you spot a node to be more central than other nodes. What is the stability of the centrality estimates?

#--> Assess stability of the centrality index by case-dropping subset bootstrap

#Centrality stability Total

set.seed(1)

CANTotal\_CentStability <- bootnet(DataCovidTotal, nBoots = 1000,

default = "mgm", type = "case",

statistics = c("Strength"), level = LevelCovidTotal, criterion="CV", binarySign=FALSE)

corStability(CANTotal\_CentStability)

pdf('S6 Centrality stability\_totalsample.pdf')

plot(CANTotal\_CentStability, subsetRange = c(100,50), statistics = c("Strength"))

dev.off()

pdf('Centrality stability\_totalsample\_order.pdf', height = 70, width = 50)

plot(CANTotal\_CentStability, "Strength", perNode = TRUE, subsetRange = c(100,50))

dev.off()

#Centrality stability UK

set.seed(1)

CANUK\_CentStability <- bootnet(DataCovidUK, nBoots = 1000, default = "mgm", type = "case",

statistics = c("Strength"), level = LevelCovidUK, criterion="CV", binarySign=FALSE)

corStability(CANUK\_CentStability)

pdf('S6 Centrality stability\_UK.pdf')

plot(CANUK\_CentStability, subsetRange = c(100,50), statistics = c("Strength"))

dev.off()

pdf('Centrality stability\_UK\_order.pdf', height = 70, width = 50)

plot(CANUK\_CentStability, "Strength", perNode = TRUE, subsetRange = c(100,50))

dev.off()

#Centrality stability NL

set.seed(1)

CANNL\_CentStability <- bootnet(DataCovidNL, nBoots = 1000, default = "mgm", type = "case",

statistics = c("Strength"), level = LevelCovidNL, criterion="CV", binarySign=FALSE)

corStability(CANNL\_CentStability)

pdf('S6 Centrality stability\_NL.pdf')

plot(CANNL\_CentStability, subsetRange = c(100,50), statistics = c("Strength"))

dev.off()

pdf('Centrality stability\_NL\_order.pdf', height = 70, width = 50)

plot(CANNL\_CentStability, "Strength", perNode = TRUE, subsetRange = c(100,50))

dev.off()

###############################

#Accuracy

##

#Say you spot a difference in edges, one thicker than the other: Is it meaningfully stronger? Check accuracy

#assess the accuracy of estimated network connections --> Obtain confidence interval around estimated edge weight using nonparametric bootstrapping

#Edge accuracy Total

set.seed(1)

Bootnet\_CANTotal<- bootnet(DataCovidTotal, nBoots = 1000,

default = "mgm", level = LevelCovidTotal, criterion="CV", binarySign=FALSE, labels = ColnamesCovidTotal)

pdf('S5 Edge accuracy\_totalsample.pdf', height = 50, width = 50)

plot(Bootnet\_CANTotal, labels = TRUE, order = "sample", res=150)

dev.off()

#Edge accuracy UK

set.seed(1)

Bootnet\_CANUK<- bootnet(DataCovidUK, nBoots = 1000,

default = "mgm", level = LevelCovidUK, criterion="CV", binarySign=FALSE, labels = ColnamesCovidUK)

pdf('S5 Edge accuracy\_UK.pdf', height = 50, width = 50)

plot(Bootnet\_CANUK, labels = TRUE, order = "sample", res=150)

dev.off()

##Edge accuracy NL

set.seed(1)

Bootnet\_CANNL<- bootnet(DataCovidNL, nBoots = 1000,

default = "mgm", level = LevelCovidNL, criterion="CV", binarySign=FALSE, labels = ColnamesCovidNL)

pdf('S5 Edge accuracy\_NL.pdf', height = 50, width = 50)

plot(Bootnet\_CANNL, labels = TRUE, order = "sample", res=150)

dev.off()

#########Difference tests

#test whether network connections and centrality estimates for different variables differ from each other

#Total

#Edge weight difference

pdf('S5 Edge difference\_totalsample.pdf', height = 50, width = 50)

plot(Bootnet\_CANTotal, statistics = "edge", plot = "difference", onlyNonZero = TRUE)

dev.off()

#Centrality difference test

pdf('S6 Centrality difference\_totalsample.pdf', height = 50, width = 50)

plot(Bootnet\_CANTotal, statistics = c("Strength"), plot ="difference")

dev.off()

#UK

#Edge weight difference test

pdf('S5 Edge difference\_UK.pdf', height = 50, width = 50)

plot(Bootnet\_CANUK, statistics = "edge", plot = "difference", onlyNonZero = TRUE)

dev.off()

#Centrality difference test

pdf('S6 Centrality difference\_UK.pdf', height = 50, width = 50)

plot(Bootnet\_CANUK, statistics = c("Strength"), plot ="difference")

dev.off()

#NL

#Edge weight difference test

pdf('S5 Edge difference\_NL.pdf', height = 50, width = 50)

plot(Bootnet\_CANNL, statistics = "edge", plot = "difference", onlyNonZero = TRUE)

dev.off()

#Centrality difference test

pdf('S6 Centrality difference\_NL.pdf', height = 50, width = 50)

plot(Bootnet\_CANNL, statistics = c("Strength"), plot ="difference")

dev.off()

########################################