if (!require(igraph)) {install.packages("igraph",repos = "http://cran.us.r-project.org"); require(effects)}

library("igraph")

if (!require(qgraph)) {install.packages("qgraph",repos = "http://cran.us.r-project.org"); require(qgraph)}

library("qgraph")

if (!require(foreign)) {install.packages("foreign",repos = "http://cran.us.r-project.org"); require(foreign)}

library("foreign")

if (!require(bootnet)) {install.packages("bootnet",repos = "http://cran.us.r-project.org"); require(bootnet)}

library("bootnet")

if (!require(dplyr)) {install.packages("dplyr",repos = "http://cran.us.r-project.org"); require(dplyr)}

library("dplyr")

if (!require(gridExtra)) {install.packages("gridExtra",repos = "http://cran.us.r-project.org"); require(gridExtra)}

library("gridExtra")

if (!require(gridBase)) {install.packages("gridBase",repos = "http://cran.us.r-project.org"); require(gridBase)}

library("gridBase")

source("http://bioconductor.org/biocLite.R")

biocLite("Rgraphviz")

if (!require(devtools)) {install.packages("devtools",repos = "http://cran.us.r-project.org"); require(devtools)}

library("devtools")

if (!require(networktools)) {install.packages("networktools",repos = "http://cran.us.r-project.org"); require(networktools)}

library("networktools")

if (!require(CTT)) {install.packages("CTT",repos = "http://cran.us.r-project.org"); require(CTT)}

library("CTT")

sessionInfo()

fulldata <- read.spss("SmilesData.sav", to.data.frame=T)

Demograph <- fulldata[!is.na(fulldata$PTGI1),]

##Demographics Descriptives

dim(Demograph)

table(Demograph$gender)

table(Demograph$race)

mean(Demograph$age, na.rm = TRUE)

sd(Demograph$age, na.rm = TRUE)

table(Demograph$cause)

table(Demograph$relat)

table(Demograph$peduc)

##Univariate Descriptives

CGitems <- Demograph[,c(paste("ICG", c(4:5, 7, 9:11, 14:15, 17, 22, 25, 26, 28), sep = ""))]

PTGitems <-Demograph[,c(paste("PTGI", c(1:2, 5, 7:8, 10:11, 18:20), sep = ""))]

colnames(CGitems)<-c("noacc", "yearn", "anger", "shock", "notrst", "nocare", "discon", "mnglss", "numb", "futmng", "identi", "wldvw", "nctrl")

colnames(PTGitems)<-c("chgpri", "applif", "spirit", "nwpth", "close", "handle", "better", "faith", "strgr", "people")

descriptCG <- cbind(apply(CGitems, 2, mean), apply(CGitems, 2, sd))

descriptPTG<-cbind(apply(PTGitems,2, mean), apply(PTGitems, 2, sd))

colnames(descriptCG)<-c("Mean", "SD")

colnames(descriptPTG) <-c("Mean", "SD")

##Reliability Analyses

itemAnalysis(CGitems) ## reliability of .91

itemAnalysis(PTGitems) ## reliability of .92

##Network analysis

###Making the data matrices

netdata3 <- fulldata[,c(paste("ICG", c(4:5, 7, 9:11, 14:15, 17, 22, 25, 26, 28), sep=""), paste("PTGI", c(1:2, 5, 7:8, 10:11, 18:20), sep=""))]

netdata3 <- na.omit(netdata3)

dim(netdata3)

netdata3<-data.matrix(netdata3)

dim(netdata3)

netdata3<-data.matrix(netdata3)

nodenamesCG <- c("noacc", "yearn", "anger", "shock", "notrst", "ncare", "discon", "mnglss", "numb", "futmng", "identi", "wldvw", "nctrl")

nodenamesPTGI <- c("chgpri", "applif", "spirit", "newpth", "close", "handle", "better", "faith", "strngr", "people")

nodenamesALL <- c(nodenamesCG, nodenamesPTGI)

nodenamesCG\_plot <- c("noacc", "yearn", "anger", "shock", "notrst", "ncare", "discon", "mnglss", "numb", "futmng", "IDENTI", "wldvw", "NCTRL")

nodenamesPTGI\_plot <- c("chgpri", "applif", "spirit", "NWPTH", "close", "handle", "better", "faith", "STRGR", "people")

nodenamesALL\_plot <- c("noacc", "yearn", "anger", "shock", "notrst", "NCARE", "discon", "mnglss", "numb", "futmng", "identi", "WLDVW", "nctrl", "chgpri", "applif", "spirit", "nwpth", "close", "handle", "better", "faith", "strgr", "people")

cg\_ptg <- netdata3[,c(paste("ICG", c(4:5, 7, 9:11, 14:15, 17, 22, 25, 26, 28), sep=""), paste("PTGI", c(1:2, 5, 7:8, 10:11, 18:20), sep=""))]

cg\_only <- netdata3[,c(paste("ICG", c(4:5, 7, 9:11, 14:15, 17, 22, 25, 26, 28), sep=""))]

ptg\_only <- netdata3[,c(paste("PTGI", c(1:2, 5, 7:8, 10:11, 18:20), sep=""))]

cg\_ptg\_d <- data.frame(cg\_ptg)

colnames(netdata3[,1:23]) <- nodenamesALL

colnames(cg\_ptg\_d) <- nodenamesALL

colnames(cg\_only) <- nodenamesCG

colnames(cg\_ptg) <- nodenamesALL

colnames(ptg\_only) <- nodenamesPTGI

###CG Network

cor\_cg\_only <- invisible(cor\_auto(cg\_only))

glasso\_cg\_only <- EBICglasso(cor\_cg\_only, n=dim(cg\_only)[1], gamma=0.5)

CGnet <- qgraph(glasso\_cg\_only, layout="spring", labels=nodenamesCG\_plot, vsize=9,

label.cex=c(rep(.7,10),.6,.7,.6),

label.font=c(rep(1,10),2,1,2),

label.color=c(rep(1, 10),"blue", 1,"blue"),

label.scale=F, DoNotPlot=F)

pdf("CGn.pdf", width = 3.8, height = 4)

CGbw<-makeBW(CGnet)

dev.off()

###CG Expected Influence

EI\_cg <- expectedInf(glasso\_cg\_only)

pdf("CGei.pdf", width = 5)

plot(EI\_cg$step1, order="value", zscore=F, yaxt = "n")

dev.off()

###Edge Weight and EI stability

set.seed(123)

net <- estimateNetwork(cg\_only, default="EBICglasso")

net\_boot <- bootnet\_flex(net, statistics=c("edge", "expectedInf"), nBoots=1000, type="case", caseN = 50)

CorStabCG<-corStability(net\_boot, statistics=c("edge", "expectedInf"))

###PTG Network

cor\_ptg\_only <- invisible(cor\_auto(ptg\_only))

glasso\_ptg\_only <- EBICglasso(cor\_ptg\_only, n=dim(ptg\_only)[1], gamma=0.5)

PTGInet<- qgraph(glasso\_ptg\_only, layout="spring", labels=nodenamesPTGI\_plot, vsize=9,

label.cex=c(rep(.7,3),.6,rep(.7, 4),.6,.7),

label.font=c(rep(1,3),2,rep(1,4),2),

label.color=c(rep(1,3),"blue", rep(1,4),"blue"),

label.scale=F, DoNotPlot=F)

PTGbw<-makeBW(PTGInet)

pdf("PTGn.pdf", width = 3.8, height = 4)

PTGbw<-makeBW(PTGInet)

dev.off()

###PTG Expected Influence

EI\_ptg <- expectedInf(glasso\_ptg\_only)

PTGIei <- plot(EI\_ptg$step1, order="value", zscore=F)

PTGIei

pdf("PTGei.pdf", width = 5)

plot(EI\_ptg$step1, order="value", zscore=F)

dev.off()

###Edge Weight and EI stability

net2 <- estimateNetwork(ptg\_only, default="EBICglasso")

net\_boot2 <- bootnet\_flex(net2, statistics=c("edge", "expectedInf"), nBoots=1000, type="case", caseN = 50)

CorStabPTG <-corStability(net\_boot2, statistics=c("edge", "expectedInf"))

###Combined Network

cor\_netCombo <- cor\_auto(cg\_ptg)

glassoCombo <- EBICglasso(cor\_netCombo, n=dim(cg\_ptg)[1], gamma=0.5)

COMBOnet <- qgraph(glassoCombo, layout="spring", labels=nodenamesALL\_plot, groups=list(CG=1:13, PTG=14:23),legend=F,

colors= c("white", "lightblue"),

label.cex=c(rep(.6, 5),.5,rep(.6,5),.5, rep(.6,11)),

label.font=c(rep(1, 5),2, rep(1,5),2, rep(1,11)),

label.color=c(rep(1, 5),"darkred",rep(1,5),"blue", rep(1,11)),

label.scale=F, DoNotPlot=F)

write.csv(cor\_netCombo, file = "S3.csv")

ComboBW <- makeBW(COMBOnet)

pdf("COMBOn.pdf", width = 5, height = 4)

COMBObw<-makeBW(COMBOnet)

dev.off()

bridgesCombo <- bridge(glassoCombo, communities=c(rep("CG", 13), rep("Growth", 10)))

BEI<-plot(bridgesCombo, order="value", zscore=F, include = "Bridge Expected Influence (1-step)")

pdf("BEI.pdf", width = 5)

plot(bridgesCombo, order="value", zscore=F, include = "Bridge Expected Influence (1-step)")

dev.off()

###spinglass community detection

spinglass.community(graph\_from\_adjacency\_matrix(glassoCombo, mode = "lower", weighted = TRUE), spins = 2)

###Edge weight stability

net3 <- estimateNetwork(cg\_ptg, default="EBICglasso")

net\_boot3 <- bootnet\_flex(net3, statistics=c("edge", "expectedInf"), nBoots=1000, type="case", caseN = 50)

CorStabPTG <-corStability(net\_boot2, statistics=c("edge", "expectedInf"))