Longitudinal Depression Supplement

## Set number of covariates  
numCovar <- 4  
## Set number of symptoms  
k <- 9  
  
## Set up empty matrix of coefficients  
adjMatCov.early <- matrix(0, (numCovar+k), (numCovar+k))   
adjMatCov.middle <- matrix(0, (numCovar+k), (numCovar+k))   
  
## Estimate CLPN for early  
for (i in 1:k) {  
 set.seed(1)  
 lassoregCov.early <- cv.glmnet(data.matrix(earlyDat[,c(1:k,(k\*2+1):(k\*2+numCovar))]), earlyDat[,(k+i)], nfolds=10,  
 family="binomial", alpha=1, standardize=TRUE)  
 lambdaCov.early <- lassoregCov.early$lambda.min  
 adjMatCov.early[(1:(k+numCovar)),i] <- coef(lassoregCov.early, s=lambdaCov.early, exact=FALSE)[2:(numCovar+k+1)]   
}  
  
## Estimate CLPN for middle  
for (i in 1:k) {  
 set.seed(1)  
 lassoregCov.middle <- cv.glmnet(data.matrix(middleDat[,c(1:k,(k\*2+1):(k\*2+numCovar))]), middleDat[,(k+i)], nfolds=10,  
 family="binomial", alpha=1, standardize=TRUE)  
 lambdaCov.middle <- lassoregCov.middle$lambda.min  
 adjMatCov.middle[(1:(k+numCovar)),i] <- coef(lassoregCov.middle, s=lambdaCov.middle, exact=FALSE)[2:(numCovar+k+1)]   
}  
  
## Adjacency matrix  
adjMatCov.early <- getWmat(adjMatCov.early, nNodes=k+NumCovar, labels=labelsCov, directed=T) # note: DVs are in columns  
adjMatCov.middle <- getWmat(adjMatCov.middle, nNodes=k+NumCovar, labels=labelsCov, directed=T) # note: DVs are in columns  
  
# Remove covariates from adjacency matrix  
adjMat.early <- adjMatCov.early[1:k, 1:k]  
adjMat.middle <- adjMatCov.middle[1:k, 1:k]  
  
## Convert from logit/log odds to odds ratios  
adjMatOR.early <- exp(adjMat.early)  
adjMatOR.middle <- exp(adjMat.middle)  
adjMatOR.early  
adjMatOR.middle

CLPN.OR <- function(data) {  
 ## create empty adjacency matrix  
 adjMatCovCLPN <- matrix(0, k+numCovar, k+numCovar)   
 ## run CLPN loop to do series of nodewise regularized regressions  
 for (i in 1:9) {  
 # set.seed(1) # commented out so that it doesn't give the same answer every time when bootstrapping  
 lassoreg <- cv.glmnet(x=data.matrix(data[,c(1:k,(k\*2+1):(k\*2+numCovar))]),   
 y=data[,(k+i)], nfolds=10, grouped=TRUE,  
 family="binomial", alpha=1, standardize=TRUE)  
 lambda <- lassoreg$lambda.min  
 ## paste coefficients into adjacency matrix  
 adjMatCovCLPN[1:(k+numCovar),i] <- coef(lassoreg, s=lambda, exact=FALSE)[2:(k+numCovar+1)]  
 }  
 ## remove covariates from adjacency matrix  
 adjMatCLPN <- adjMatCovCLPN[1:k, 1:k]  
 # Convert from logit/log odds to odds ratios  
 adjMatCLPN.OR <- exp(adjMatCLPN)  
 return(adjMatCLPN.OR)  
 #return(adjMatCLPN)  
}  
  
## Estimate networks in bootnet (for bootstrapping)  
set.seed(1)  
net.early <- estimateNetwork(earlyDat, fun=CLPN.OR, labels=labels, directed=T)  
set.seed(1)  
net.middle <- estimateNetwork(middleDat, fun=CLPN.OR, labels=labels, directed=T)

set.seed(1)  
nonParBoot.early.or <- bootnet(net.early, type="nonparametric", nBoots=5000, directed=T,  
 statistics=c("edge"),  
 ncores=8)  
  
set.seed(1)  
nonParBoot.middle.or <- bootnet(net.middle, type="nonparametric", nBoots=5000, directed=T,  
 statistics=c("edge"),  
 ncores=8)

#estimate proportion of time bootstrapped edge > 1  
#this is because the 95% CI's don't really reflect confidence intervals  
#see https://psych-networks.com/bootstrapping-edges-after-regularization-clarifications-tutorial/  
early.bootTable <-nonParBoot.early.or$bootTable  
early\_edge\_conf <-early.bootTable %>% group\_by(id) %>%   
 mutate(total\_count = n(),  
 conf = ifelse(value > 1 | value < 1, 1, 0)) %>%  
 summarize(conf\_count = sum(conf),  
 Confidence = conf\_count/total\_count) %>%  
 distinct(id, .keep\_all = T) %>%  
 separate(id, c("nodeOut", "nodeIn"), sep="->") %>% select(-conf\_count)

## `summarise()` has grouped output by 'id'. You can override using the `.groups` argument.

middle.bootTable <-nonParBoot.middle.or$bootTable  
middle\_edge\_conf <- middle.bootTable %>% group\_by(id) %>%   
 mutate(total\_count = n(),  
 conf = ifelse(value > 1| value < 1, 1, 0)) %>%  
 summarize(conf\_count = sum(conf),  
 Confidence = conf\_count/total\_count) %>%  
 distinct(id, .keep\_all = T) %>%  
 separate(id, c("nodeOut", "nodeIn"), sep="->") %>% select(-conf\_count)

## `summarise()` has grouped output by 'id'. You can override using the `.groups` argument.

## Identify the strongest edges (note: row=IV, col=DV)  
res.early <- order(adjMatOR.early, decreasing = T)[seq\_len(81)]  
pos.early <- arrayInd(res.early, dim(adjMatOR.early), useNames = TRUE)  
posWithLabs.early <- data.frame(nodeOut=labels[pos.early[,1]],   
 nodeIn=labels[pos.early[,2]],  
 value=adjMatOR.early[res.early])  
Early.edges<-posWithLabs.early %>% mutate(Odds\_Ratio = value) %>%  
 left\_join(early\_edge\_conf, by=c("nodeOut", "nodeIn"))%>% drop\_na() %>% filter(Odds\_Ratio > 1 | Odds\_Ratio < 1) #drop\_na omits the autoregressive edges  
  
  
res.middle <- order(adjMatOR.middle, decreasing = T)[seq\_len(81)]  
pos.middle <- arrayInd(res.middle, dim(adjMatOR.middle), useNames = TRUE)  
posWithLabs.middle <- data.frame(nodeOut=labelsCov[1:9][pos.middle[,1]],  
 nodeIn=labelsCov[1:9][pos.middle[,2]],  
 value=adjMatOR.middle[res.middle])  
Middle.edges<-posWithLabs.middle %>% mutate(Odds\_Ratio = value) %>% select(-c(value)) %>%  
 left\_join(middle\_edge\_conf, by=c("nodeOut", "nodeIn")) %>% drop\_na()%>% filter(Odds\_Ratio > 1 | Odds\_Ratio < 1)  
  
#Table 2 in Manuscript  
joined\_table <- left\_join(Early.edges, Middle.edges, by=c("nodeOut", "nodeIn"))  
colnames(joined\_table) <- c("nodeOut", "nodeIn", "Odds\_Ratio\_Early", "Confidence\_Early", "Odds\_Ratio\_Middle", "Confidence\_Middle")  
joined\_table

## nodeOut nodeIn Odds\_Ratio\_Early Confidence\_Early Odds\_Ratio\_Middle  
## 1 failure SI 3.674514 3.674514 0.9330  
## 2 appetite SI 2.844684 2.844684 0.9260  
## 3 energy sleep 2.118878 2.118878 0.9862  
## 4 failure energy 1.894185 1.894185 0.8316  
## 5 energy appetite 1.867852 1.867852 0.8922  
## 6 failure sleep 1.579375 1.579375 0.7482  
## 7 energy failure 1.381765 1.381765 0.7360  
## 8 appetite failure 1.313198 1.313198 0.6766  
## 9 sleep SI 1.277288 1.277288 0.5924  
## 10 sleep appetite 1.223639 1.223639 0.5598  
## 11 failure appetite 1.054192 1.054192 0.4014  
## Confidence\_Middle NA  
## 1 6.023010 0.8694  
## 2 2.684947 0.8320  
## 3 NA NA  
## 4 NA NA  
## 5 1.120547 0.6548  
## 6 1.032745 0.5088  
## 7 NA NA  
## 8 NA NA  
## 9 NA NA  
## 10 1.307258 0.7892  
## 11 1.929518 0.8846

#models not run to generate the data for the supplement - we read the saved data in  
set.seed(1)  
caseBoot.early <- bootnet(net.early, type="case", nBoots=5000, directed=T, caseN = 50,  
 statistics=c( "outExpectedInfluence","inExpectedInfluence"), ncores=8)  
  
set.seed(1)  
caseBoot.middle <- bootnet(net.middle, type="case", nBoots=5000, directed=T, caseN = 50,  
 statistics=c( "outExpectedInfluence","inExpectedInfluence"), ncores=8)

## Compute CS-coefficients  
corStability(caseBoot.early, cor=.7)

## === Correlation Stability Analysis ===   
##   
## Sampling levels tested:  
## nPerson Drop% n  
## 1 77 75.1 77  
## 2 82 73.5 85  
## 3 86 72.2 92  
## 4 90 70.9 80  
## 5 95 69.3 87  
## 6 99 68.0 88  
## 7 104 66.3 98  
## 8 108 65.0 98  
## 9 113 63.4 94  
## 10 117 62.1 93  
## 11 121 60.8 93  
## 12 126 59.2 102  
## 13 130 57.9 105  
## 14 135 56.3 106  
## 15 139 55.0 112  
## 16 143 53.7 83  
## 17 148 52.1 95  
## 18 152 50.8 107  
## 19 157 49.2 109  
## 20 161 47.9 95  
## 21 166 46.3 95  
## 22 170 45.0 95  
## 23 174 43.7 119  
## 24 179 42.1 103  
## 25 183 40.8 96  
## 26 188 39.2 103  
## 27 192 37.9 104  
## 28 196 36.6 96  
## 29 201 35.0 88  
## 30 205 33.7 88  
## 31 210 32.0 103  
## 32 214 30.7 103  
## 33 219 29.1 111  
## 34 223 27.8 99  
## 35 227 26.5 102  
## 36 232 24.9 109  
## 37 236 23.6 108  
## 38 241 22.0 99  
## 39 245 20.7 105  
## 40 249 19.4 102  
## 41 254 17.8 77  
## 42 258 16.5 103  
## 43 263 14.9 109  
## 44 267 13.6 106  
## 45 271 12.3 99  
## 46 276 10.7 104  
## 47 280 9.4 118  
## 48 285 7.8 113  
## 49 289 6.5 127  
## 50 294 4.9 117  
##   
## Maximum drop proportions to retain correlation of 0.7 in at least 95% of the samples:  
##   
## hybrid: 0   
## - For more accuracy, run bootnet(..., caseMin = 0, caseMax = 0.049)   
##   
## inExpectedInfluence: 0.107   
## - For more accuracy, run bootnet(..., caseMin = 0.094, caseMax = 0.123)   
##   
## inStrength: 0.107   
## - For more accuracy, run bootnet(..., caseMin = 0.094, caseMax = 0.123)   
##   
## outExpectedInfluence: 0.22   
## - For more accuracy, run bootnet(..., caseMin = 0.207, caseMax = 0.236)   
##   
## outStrength: 0.22   
## - For more accuracy, run bootnet(..., caseMin = 0.207, caseMax = 0.236)   
##   
## rspbc: 0   
## - For more accuracy, run bootnet(..., caseMin = 0, caseMax = 0.049)   
##   
## Accuracy can also be increased by increasing both 'nBoots' and 'caseN'.

corStability(caseBoot.middle, cor=.7)

## === Correlation Stability Analysis ===   
##   
## Sampling levels tested:  
## nPerson Drop% n  
## 1 64 74.9 55  
## 2 67 73.7 64  
## 3 71 72.2 70  
## 4 75 70.6 64  
## 5 78 69.4 70  
## 6 82 67.8 77  
## 7 86 66.3 87  
## 8 89 65.1 86  
## 9 93 63.5 76  
## 10 97 62.0 88  
## 11 100 60.8 95  
## 12 104 59.2 81  
## 13 107 58.0 111  
## 14 111 56.5 103  
## 15 115 54.9 103  
## 16 118 53.7 97  
## 17 122 52.2 125  
## 18 126 50.6 105  
## 19 129 49.4 100  
## 20 133 47.8 101  
## 21 137 46.3 104  
## 22 140 45.1 99  
## 23 144 43.5 106  
## 24 148 42.0 108  
## 25 151 40.8 123  
## 26 155 39.2 98  
## 27 158 38.0 103  
## 28 162 36.5 126  
## 29 166 34.9 94  
## 30 169 33.7 112  
## 31 173 32.2 109  
## 32 177 30.6 101  
## 33 180 29.4 83  
## 34 184 27.8 136  
## 35 188 26.3 102  
## 36 191 25.1 132  
## 37 195 23.5 101  
## 38 199 22.0 112  
## 39 202 20.8 103  
## 40 206 19.2 106  
## 41 209 18.0 98  
## 42 213 16.5 90  
## 43 217 14.9 112  
## 44 220 13.7 113  
## 45 224 12.2 105  
## 46 228 10.6 110  
## 47 231 9.4 118  
## 48 235 7.8 106  
## 49 239 6.3 111  
## 50 242 5.1 121  
##   
## Maximum drop proportions to retain correlation of 0.7 in at least 95% of the samples:  
##   
## hybrid: 0   
## - For more accuracy, run bootnet(..., caseMin = 0, caseMax = 0.051)   
##   
## inExpectedInfluence: 0   
## - For more accuracy, run bootnet(..., caseMin = 0, caseMax = 0.051)   
##   
## inStrength: 0   
## - For more accuracy, run bootnet(..., caseMin = 0, caseMax = 0.051)   
##   
## outExpectedInfluence: 0.078   
## - For more accuracy, run bootnet(..., caseMin = 0.063, caseMax = 0.094)   
##   
## outStrength: 0.078   
## - For more accuracy, run bootnet(..., caseMin = 0.063, caseMax = 0.094)   
##   
## Accuracy can also be increased by increasing both 'nBoots' and 'caseN'.

## Coefficient of similarity  
cor.test(c(adjMatOR.early[lower.tri(adjMatOR.early, diag=F)], adjMatOR.early[upper.tri(adjMatOR.early, diag=F)]),   
 c(adjMatOR.middle[lower.tri(adjMatOR.middle, diag=F)], adjMatOR.middle[upper.tri(adjMatOR.middle, diag=F)]),   
 method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: c(adjMatOR.early[lower.tri(adjMatOR.early, diag = F)], adjMatOR.early[upper.tri(adjMatOR.early, diag = F)]) and c(adjMatOR.middle[lower.tri(adjMatOR.middle, diag = F)], adjMatOR.middle[upper.tri(adjMatOR.middle, diag = F)])  
## t = 0.86958, df = 70, p-value = 0.3875  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.1314383 0.3272102  
## sample estimates:  
## cor   
## 0.1033781

## No. (%) of edges in early network that replicated (i.e., present with same sign) in middle network  
paste0(sum(adjMatOR.early>1 & adjMatOR.middle>1) + sum(adjMatOR.early<1 & adjMatOR.middle<1), " (",   
 round(100\*((sum(adjMatOR.early>1 & adjMatOR.middle>1) + sum(adjMatOR.early<1 & adjMatOR.middle<1)) / sum(adjMatOR.early!=1)),1), "%)")

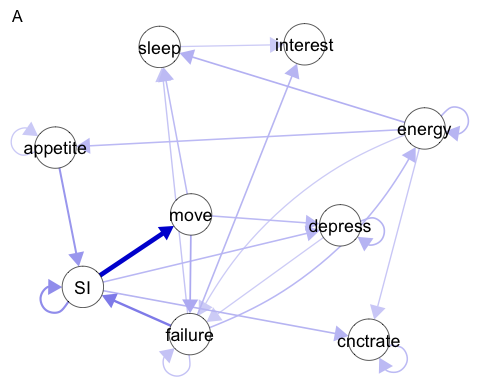
## [1] "25 (59.5%)"

## Jaccard Index (edges present with same sign in both networks / all edges in either network)  
(sum(adjMatOR.early>1 & adjMatOR.middle>1) +   
 sum(adjMatOR.early<1 & adjMatOR.middle<1)) /   
 sum(adjMatOR.early!=1 | adjMatOR.middle!=1)

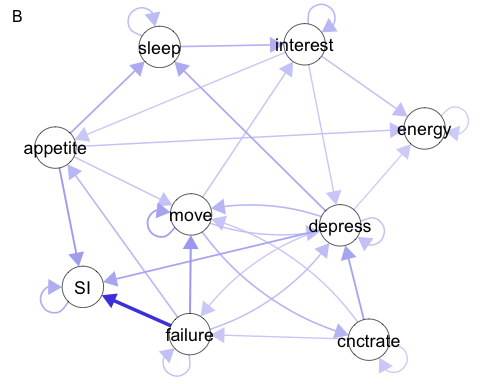
## [1] 0.3787879

maxEdge <- max(c(adjMatOR.early, adjMatOR.middle)) # 1.264336  
layoutwoAR <- averageLayout(adjMatOR.early, adjMatOR.middle)  
  
#layout(matrix(c(1,2), nrow=1), widths=c(2,2), heights=2)

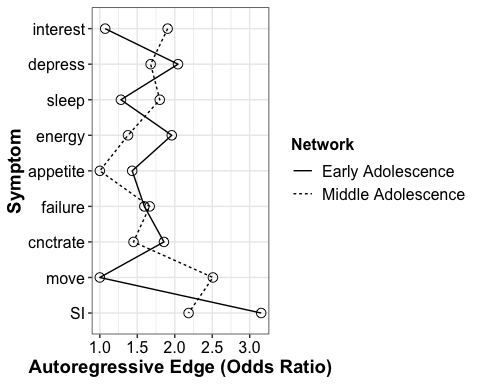
plotwoAR.early <- qgraph(adjMatOR.early, labels=labels, layout=layoutwoAR,  
 legend=F, nodeNames=labels, maximum=maxEdge, asize=7,  
 vsize=9, label.cex=2, theme="colorblind", threshold=1.35, title="A",  
 label.scale.equal = T)



plotwoAR.middle <- qgraph(adjMatOR.middle, labels=labels, layout=layoutwoAR,  
 legend=F, nodeNames=labels, maximum=maxEdge, asize=7,  
 vsize=9, label.cex=2, theme="colorblind", threshold=1.35, title="B",  
 label.scale.equal = T)

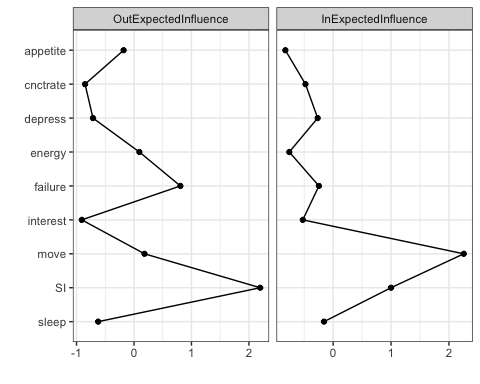


## Save the autoregressive edges  
k <- 9  
ARedges.early <- diag(adjMatOR.early)  
ARedges.middle <- diag(adjMatOR.middle)  
  
## Make df for ggplot  
ARedges <- data.frame(Symptom=rep(factor(shortnames,   
 levels=rev(shortnames)), 2),  
 Network=c(rep("Early Adolescence", k),  
 rep("Middle Adolescence", k)),  
 ARedge=c(ARedges.early,   
 ARedges.middle)  
)  
ARedges$Network <- factor(ARedges$Network, levels=c("Early Adolescence",  
 "Middle Adolescence"))  
  
## Plot autoregressive edges as a line graph (Figure S1)  
ggplot(ARedges, aes(x=Symptom, y=ARedge, group=Network)) +  
 geom\_path(aes(lty=Network)) +  
 geom\_point(aes(group=Network), shape=21, size=3) +  
 coord\_flip() +  
 ylab("Autoregressive Edge (Odds Ratio)") + xlab("Symptom") +   
 scale\_fill\_manual(values=c('#E69F00', '#56B4E9', '#009E73')) +  
 scale\_color\_manual(values=c('#E69F00', '#56B4E9', '#009E73')) +  
 guides(lty=guide\_legend(order=1),  
 fill=guide\_legend(order=2),   
 col=FALSE) +  
 theme\_bw() +  
 theme(text=element\_text(color="black"),  
 plot.title=element\_text(size=14, color="black", hjust=.5, face="bold"),  
 axis.text=element\_text(size=12, color="black"),  
 axis.title=element\_text(size=14, color="black", face="bold"),  
 legend.title=element\_text(size=12, face="bold"),  
 legend.text=element\_text(size=12, color="black"))



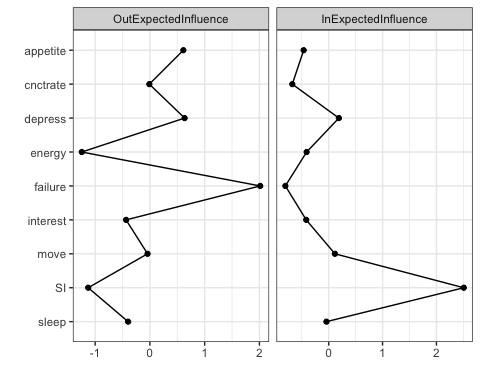
##centrality plots (only for supplement)  
## Centrality plot  
centPlot.early <- centralityPlot(adjMatOR.early,   
 include=c("OutExpectedInfluence","InExpectedInfluence"),  
 labels=labelsCov[1:9], scale="z-scores",  
 decreasing=T)

## Note: z-scores are shown on x-axis rather than raw centrality indices.



centPlot.middle <- centralityPlot(adjMatOR.middle,   
 include=c("OutExpectedInfluence","InExpectedInfluence"),  
 labels=labelsCov[1:9], scale="z-scores",  
 decreasing=T)

## Note: z-scores are shown on x-axis rather than raw centrality indices.



centPlotdf.early <- centPlot.early$data  
centPlotdf.middle <- centPlot.middle$data  
#just combine early & middle to compare in vs out between..  
cent.in.early <- centPlotdf.early %>% filter(measure=="InExpectedInfluence") %>% mutate(grp = "Early Adolescence")  
cent.in.middle <- centPlotdf.middle %>% filter(measure=="InExpectedInfluence") %>% mutate(grp = "Middle Adolescence")  
cent.in <- rbind(cent.in.early, cent.in.middle)  
## Plot in-centrality as a line graph (Figure S1)  
in\_centrality<-ggplot(cent.in, aes(x=node, y=value, group=grp)) +  
 geom\_path(aes(lty=grp)) +  
 geom\_point(aes(group=grp), shape=21, size=3) +  
 coord\_flip() +  
 ylab("z-score") + xlab("Symptom") +   
 scale\_fill\_manual(values=c('#E69F00', '#56B4E9', '#009E73')) +  
 scale\_color\_manual(values=c('#E69F00', '#56B4E9', '#009E73')) +  
 scale\_y\_continuous(expand = c(0, 0), limits = c(-1,3)) + # to make bars start at 0  
 guides(lty=guide\_legend(order=1), # set the order of the legends  
 fill=guide\_legend(order=2),   
 col=FALSE) +  
 theme\_bw() +  
 theme(text=element\_text(color="black"),  
 plot.title=element\_text(size=14, color="black", hjust=.5, face="bold"),  
 axis.text=element\_text(size=12, color="black"),  
 axis.title=element\_text(size=14, color="black", face="bold"),  
 legend.title=element\_blank(),  
 legend.text=element\_text(size=12, color="black"))  
  
cent.out.early <- centPlotdf.early %>% filter(measure=="OutExpectedInfluence") %>% mutate(grp = "Early Adolescence")  
cent.out.middle <- centPlotdf.middle %>% filter(measure=="OutExpectedInfluence") %>% mutate(grp = "Middle Adolescence")  
cent.out <- rbind(cent.out.early, cent.out.middle)  
## Plot in-centrality as a line graph (Figure S1)  
out\_centrality<-ggplot(cent.out, aes(x=node, y=value, group=grp)) +  
 geom\_path(aes(lty=grp)) +  
 geom\_point(aes(group=grp), shape=21, size=3) +  
 coord\_flip() +  
 ylab("z-score") + xlab("Symptom") +   
 scale\_fill\_manual(values=c('#E69F00', '#56B4E9', '#009E73')) +  
 scale\_color\_manual(values=c('#E69F00', '#56B4E9', '#009E73')) +  
 scale\_y\_continuous(expand = c(0, 0), limits = c(-2,3)) + # to make bars start at 0  
 guides(lty=guide\_legend(order=1), # set the order of the legends  
 fill=guide\_legend(order=2),   
 col=FALSE) +  
 theme\_bw() +  
 theme(text=element\_text(color="black"),  
 plot.title=element\_text(size=14, color="black", hjust=.5, face="bold"),  
 axis.text=element\_text(size=12, color="black"),  
 axis.title=element\_text(size=14, color="black", face="bold"),  
 legend.title=element\_blank(),  
 legend.text=element\_text(size=12, color="black"))  
  
cowplot::plot\_grid(in\_centrality, out\_centrality, labels = c("IN", "OUT"))

