**Supplementary Materials**

**S1. Network accuracy and stability**

**Method:** the accuracy tests of network indices were performed in three steps: 1) estimating edge weight accuracy by drawing bootstrapped Confidence Intervals (CIs); 2) investigating the stability of the centrality indices using CS-coefficients; 3) performing bootstrapped difference tests between edge weights and centrality indices and constructing a box-plot (Epskamp et al., 2017).

**Results:** As illustrated in*Supplementary Figure 1*, nodes were bootstrapped to test the stability of centrality, which showed sharp droppings of closeness and betweenness, but moderately stable strength centrality. The bootstrapped difference test was also used to test if edge-weights and centrality indices significantly differed from each other. Results are shown in *Supplementary Figure 2*. The edge weight accuracy test revealed relatively small CIs, suggesting a relatively precise estimation. The edge weight difference test revealed that our network was accurately estimated and that the strongest edges were significantly stronger than other edges. The stability of centrality estimates was quantified by the correlation-stability-coefficient (CS Coefficients). In the current estimation, CS coefficients were 0.67 for strength, 0.28 for closeness, and 0.05 for betweenness. As the main indicator, strength centrality CS coefficient was above 0.5, suggesting stability of the model. However, the betweenness index in our network was below 0.25. *Supplementary Figure 3* shows results from the bootstrapped difference test on node strength of the 10 subscales, which featured cogent differences in the TAS-F1, MIS and CSAS subscales, indicating they differed most strongly from other nodes in the network.



**Supplementary Figure 1. Average correlations between centrality indices of networks sampled with cases dropped.** Lines indicate the means and areas indicate the range from the 2.5th quantile to the 97.5 quantile.



**Supplementary Figure 2. Bootstrapped confidence intervals of estimated edge-weights for the estimated network of 10 subscales.**

The red line indicates the sample values and the gray area the bootstrapped CIs. Each horizontal line represents one edge of the network, ordered from the edge with the highest edge-weight to the edge with the lowest edge weight. In the case of ties (multiple edges-weights were estimated to be exact 0), the mean of the bootstrap samples was used in ordering the edges. The y-axis labels have been removed to avoid cluttering.

****

**Supplementary Figure 3. Bootstrapped difference test (α=0.05) node strength of the 10 subscales.** Grey boxes indicate nodes that do not differ significantly from one-another and black boxes represent nodes that do differ significantly from one-another. White boxes show the value of node strength.

**S3. R codes for network analysis**

**### install packages**

install.packages("qgraph") # estimate network

install.packages("bootnet") # network stability and accuracy

install.packages("mgm") # predictability

install.packages("networktools") # EI, expected influence

**# estimate network**

library("qgraph")

corMat <- cor\_auto(data)

pdf("Fig1\_final.pdf",width=8,height=8)

graph\_lasso <- qgraph(corMat, graph = "glasso", layout = "spring", tuning = 0.25,

 sampleSize = nrow(data))

dev.off()

**# centrality**

centrality(graph\_lasso)

centralityTable(network, standardized = FALSE)

pdf("Fig2\_final.pdf")

centralityPlot(graph\_lasso, scale="z-scores", include=c("Strength","Closeness","Betweenness","ExpectedInfluence"))

dev.off()

**# expected influence**

library("networktools")

EI=expectedInf(graph\_lasso, step = c("both", 1, 2), directed = FALSE)

EI$step1

EI$step2

**# predictability**

library("mgm")

data <- na.omit(data)

data <- as.matrix(data)

p <- ncol(data)

dim(data)

set.seed(1)

fit\_obj <- mgm(data = data,

 type = rep('g', p),

 level = rep(1, p),

 lambdaSel = 'CV',

 ruleReg = 'OR',

 pbar = TRUE)

pred\_obj <- predict(object = fit\_obj,

 data = data,

 errorCon = 'R2')

pred\_obj$error

pdf("Fig3\_network\_with\_predictability.pdf",width=8,height=8)

qgraph(fit\_obj$pairwise$wadj, # weighted adjacency matrix as input

 layout = 'spring',

 pie = pred\_obj$error[,2], # provide errors as input

 pieColor = rep('#377EB8',p),

 edge.color = fit\_obj$pairwise$edgecolor,

 labels = colnames(data))

dev.off()

**# Stability estimation**

library("bootnet")

network <- estimateNetwork(data, default = "EBICglasso")

boot1=bootnet(network) # bootstrapping-edge-weight-accuracy

plot(boot1, labels = FALSE, order = "sample")

boot2=bootnet(network, type = "case") # centrality\_stablity

plot(boot2)

corStability(boot2) # C-S Coefficients

differenceTest(boot1, 3, 1, "strength") # testing for significant differences

plot(boot1, "strength")