# B. R Scripts used to run co-occurrence analysis and network construction

In order to construct the networks in this study, there are three functions:

1. compute correlation coefficient all pair of variables,
2. contruct network from step1,
3. detect communities from step2.

Another function is used to visualize node properties of resulting networks to determine the functions each node has in the network.

## Libraries

if (!require("igraph")) {  
 install.packages("igraph", dependencies = TRUE)  
}  
  
if (!require("qgraph")) {  
 install.packages("qgraph", dependencies = TRUE)  
}  
  
if(!require("ggplot2")) {  
 install.packages("ggplot2", dependencies = TRUE)  
}  
  
if(!require("dplyr")) {  
 install.packages("dplyr", dependencies = TRUE)  
}  
  
if(!require("RColorBrewer")) {  
 install.packages("RColorBrewer", dependencies = TRUE)  
}

## Step 1

The step in this analysis is to perform Spearman correlation with each dataset. The computation starts at the first column containing injury values. The output is a data frame with five columns for each dataset labeled pair of injuries (2 columns), the correlation coefficient, *p*-value of the pair.

cooc\_table <- function(data){  
   
 # create the object to store the result  
 results <- matrix(nrow = 0, ncol = 6)  
   
 for (a in 1:(length(names(data)) - 1)) {  
   
 # every variable will be compared to every other variable, so there has  
 # to be another loop that iterates down the rest of the columns  
   
 for (b in (a + 1):length(names(data))) {  
   
 # summing the abundances of species of the columns that will be  
 # compared  
   
   
 var1.sum <- sum(temp[, a], na.rm = TRUE)  
 var2.sum <- sum(temp[, b], na.rm = TRUE)  
   
 # if the column is all 0's no co-occurrence will be performed  
   
 if (var1.sum > 1 & var2.sum > 1) {  
   
 # compute correlation using Spearman's correlation measure  
 # can change correlation methods based on one's objective  
 test <- cor.test(data[, a], data[, b], method = "spearman",  
 na.action = "na.exclude", exact = FALSE)  
   
 # There are warnings when setting exact = TRUE because of ties from the  
 # output of Spearman's correlation  
 # stackoverflow.com/questions/10711395/spear-man-correlation and ties  
 # It would be still valid if the data is not normally distributed.  
   
 # extract rho value  
 rho <- test$estimate  
 # extract p.value value  
 p.value <- test$p.value  
 }  
   
 if (var1.sum <= 1 | var2.sum <= 1) {  
 # if varible is too small, it will not be included in analysis  
 rho <- 0  
 p.value <- 1  
 }  
   
 # combine the value  
 new.row <- c(names(data)[a], names(data)[b], rho, p.value, var1.sum,  
 var2.sum)  
   
 # combine row of the result  
 results <- rbind(results, new.row)  
   
 }  
   
 }  
 # change the class to dataframe  
 results <- as.data.frame(results)  
   
 row.names(results) <- NULL  
   
 # rename column  
 names(results) <- c("var1", "var2", "rho", "p.value", "var1.sum", "var2.sum")  
   
 # set class of rho data to be numeric  
 results$rho <- as.numeric(as.character(results$rho))  
   
 # set class of p.value to be numberic  
 results$p.value <- as.numeric(as.character(results$p.value))  
  
 return(results)  
}

The second step is to construct a network of co-occurrence relationships in a data frame format that can be used with the igraph package to create networks. The input file contains the results from the step. The output from this is an igraph object with all edges in the network containing the Spearman correlation coefficient significant at *p*-values < 0.05.

plot\_network <- function(data){  
   
 #== adjust the vertices properties  
   
 # creat a network from an edge list matrix  
   
 net <- graph.edgelist(as.matrix(data[ ,c("var1","var2")]), directed = FALSE)  
   
 # set color's node  
 V(net)$color <- adjustcolor("khaki2", alpha.f = .8)  
 # set color's frame node  
 V(net)$frame.color <- adjustcolor("khaki2", alpha.f = .8)  
   
 # set shape of node  
 V(net)$shape <- "circle"  
   
 # set size of node. 25 is suitable for showing grap on A4 paper  
 V(net)$size <- 25  
   
 V(net)$label.color <- "black"  
   
 V(net)$label.font <- 2  
   
 V(net)$label.family <- "Helvetica"  
   
 V(net)$label.cex <- 1.0  
   
 # == adjust the edge proterties  
   
 # label weight of each node pair with correlation cofficient  
 E(net)$weight <- as.matrix(table[, "rho"])  
   
 E(net)$width <- abs(E(net)$weight)\*10  
   
 E(net)$color <- "steelblue2"  
   
 # set graph layout to Fruchterman-Reingold layout  
 net$layout <- layout\_with\_fr(net)  
   
 # output is igraph object  
 return(net)  
}

The third step takes input from the second step and generates an igraph object, which can be used with the igraph package to visualize the network with community detection using the optimal\_cluster function from the igraph package.

cluster.network <- function(graph){  
   
 # detect comminities with maximizing mudularity  
 community <- cluster\_optimal(graph, weights = abs(E(graph)$weight))  
   
 # set colors for each community  
 prettyColors <- brewer.pal(n = 8, name = 'Set2')  
   
 # define color to the community  
 V(graph)$color <- prettyColors[membership(community)]  
   
 # set the size of netwrok  
 V(graph)$size <- 15  
   
 # set width of edge  
 E(graph)$width <- abs(E(graph)$weight)\*10  
   
 return(graph)  
}

The fourth and final step is the function, which is applied for extracting node properties from each node in a network and plot those node properties with dot graph with ggplot2 package. The input file is an igraph object from the second step. The output is a ggplot2 graph showing node degree, betweenness, clustering coefficient, clustering coefficient from the network.

plot.node.centrality <- function(graph){  
   
 # call function from qgraph  
 cen <- centrality\_auto(graph)$node.centrality  
 cen$node <- row.names(cen)  
 # compute clustering coeeficient from igraph package  
 cen$CC <- igraph::transitivity(graph, type = "local", isolates = "zero")  
 row.names(cen) <- NULL  
   
 cus\_theme <- theme\_bw() + theme(  
 panel.grid.major.x = element\_blank(),  
 panel.grid.minor.x = element\_blank(),  
 panel.grid.major.y = element\_line(color = "grey60", linetype = 3, size = 0.5),  
 axis.title = element\_text(size = 18),  
 axis.text = element\_text(size = 14))  
   
 p1 <- cen %>% ggplot(aes(x= Degree, y = reorder(node, Degree))) +  
 geom\_point(size = 5, color ="red") +  
 cus\_theme +   
 xlab("Node degree") +   
 ylab("Node")   
   
 p2 <- cen %>% ggplot(aes(x= CC, y = reorder(node, CC))) +   
 geom\_point(size = 5, color ="blue") +  
 cus\_theme +theme(axis.title.y = element\_blank()) +  
 xlab("Clustering Coef")   
   
 p3 <- cen %>% ggplot(aes(x= Betweenness, y = reorder(node, Betweenness))) +   
 geom\_point(size = 5, color = "black") +  
 cus\_theme + theme(axis.title.y = element\_blank()) +  
 xlab("Betweenness")  
   
 plot\_grid(p1, p2, p3, labels=c("A", "B", "C"), ncol = 3, nrow = 1 )  
   
}