

ABIDEI PINT FC mat stats

Erin W Dickie, Saba Shahab

January 15, 2018

This is an analysis of the FC Mats

```
library(dplyr)
library(tidyr)
library(broom)
library(igraph)
library(corrplot)
library(effsize)
library(knitr)
library(ggplot2)
library(gridExtra)
```

Reading in the data

A couple things to note here about the how the clinical data is being parsed:

1. all categorical variables are being set to Factors
 - Note that sex.x still needs descriptive labels
2. Education is being very basically imputed when it is not available so that we don't lose data

```
## Modified by Saba Shahab on November 9, 2016 for SPINS-CMH data.
```

```
## set pint_dir to be the pint outputs
pint_dir <- "../data/PINT_outputs_s8_6-6-12/"

## reading in the qcdd_sublists csv to get the sublists
qcdd_sublists <- read.csv("../phenotypic/NEWallSubjects_completeData3_DM_not_sexmatched.csv")
pheno <- qcdd_sublists %>%
  mutate(Site = factor(site, levels = c(1,2,3),
    labels = c("CMH", "ZHH", "COBRE")),
    DX = factor(DX_GROUP, level = c(1,2), labels = c('SSD', 'Ctrl')),
    Sex = factor(sex.x),
    Edu = if_else(is.na(educationCode),
      true = mean(educationCode, na.rm = F),
      false = educationCode))
```

```
##load the more descriptive list of the vertices...not we may want to change a few based on these results
Yeo7_80verts <- read.csv("~/code/ciftify/ciftify/data/PINT/Yeo7_2011_80verts.csv")
```

Some functions we're using

```
## functions
read_subject_means <- function(subid, pint_dir, vertex_type) {
  means <- read.csv(file.path(pint_dir,
```

```

        subid,
        paste(subid, vertex_type, "meants.csv", sep="_"),
        header=FALSE)
  return(meants)
}

load_all_subject_matrices <- function(subids, myedgenames, pint_dir,
                                     vertex_type) {
  ## use these parameters to set up a blank dataframe to hold all the correlations
  theZs <- as.data.frame(cbind("subid" = subids,
                              matrix(numeric(),
                                    nrow = length(subids),
                                    ncol = numedges,
                                    dimnames = list(1:length(subids),
                                                    myedgenames))))
  theZs[,2:ncol(theZs)] <- numeric(nrow(theZs)*(ncol(theZs)-1))

  ## now correlate everyone's ts and write the correlations to a massive data frame
  for (i in 1:nrow(theZs)) {
    ## get the subid from the dataframe and read in the meants
    meants <- read_subject_meants(theZs$subid[i], pint_dir, vertex_type)

    ## correlate and graph
    cormat <- cor(t(meants))
    g<-graph_from_adjacency_matrix(cormat,mode="upper",
                                  weighted=T, diag=F)

    # take the edge list as a vector
    thecorrs <- E(g)$weight

    # apply the Z transform (so we can do stats)
    theseZ <- atanh(thecorrs)

    # save the output to the data.frame
    theZs[i,2:ncol(theZs)] <- theseZ
  }
  return(theZs)
}

make_corrplot <-function(data, attrtoplot, graph_title) {
  forplot <- as.data.frame(data)
  forplot <- forplot %>% separate(Edge, into = c("V1", "V2"), sep = '\\.')
  g <- graph_from_data_frame(forplot, directed=F)
  g.mat <- get.adjacency(g, attr=attrtoplot)
  corrplot(as.matrix(g.mat),
           is.corr = F, tl.cex = 0.3, tl.col = "black",
           title = graph_title)
}

#new circle graph function
radian.rescale <- function(x, start=0, direction=1) {
  c.rotate <- function(x) (x + start) %%(2 * pi) * direction
  c.rotate(scales::rescale(x, c(0, 2 * pi), range(x)))
}

```

```

}

draw_circle_graph <- function(data, Yeo7_80verts) {
  CircleOrder <- as.character(Yeo7_80verts$SHORTNAME)
  CircleOrder <- c(CircleOrder[75:length(CircleOrder)], CircleOrder[1:74])
  g <- graph.data.frame(data, directed = F)
  Yeo7_80verts$wnetname <- substr(Yeo7_80verts$SHORTNAME,3,6)
  Yeo7_80verts$Vhex <- NA
  Yeo7_80verts$Vhex[Yeo7_80verts$NETWORK==2] <- "#781286"
  Yeo7_80verts$Vhex[Yeo7_80verts$NETWORK==3] <- "#00760E"
  Yeo7_80verts$Vhex[Yeo7_80verts$NETWORK==4] <- "#C43AFA"
  Yeo7_80verts$Vhex[Yeo7_80verts$NETWORK==5] <- "#4682B4"
  Yeo7_80verts$Vhex[Yeo7_80verts$NETWORK==6] <- "#E69422"
  Yeo7_80verts$Vhex[Yeo7_80verts$NETWORK==7] <- "#CD3E3A"
  V(g)$color <- Yeo7_80verts$Vhex
  g$layout <- layout_in_circle(g,order = CircleOrder)
  lab.locs <- radian.rescale(x=1:80, direction=-1, start=0)
  d <- as.matrix(dist(g$layout, method = "euclidean",
    upper = TRUE, diag = TRUE, p = 2))
  gdlist <- graph_from_adjacency_matrix(as.matrix(dist(g$layout, method = "euclidean",
    upper = TRUE, diag = TRUE, p = 2)),
    mode="upper", weighted=T, diag=F)
  dist.df <- cbind(data,E(gdlist)$weight)
  dist.df$scaledD <- (dist.df$`E(gdlist)$weight`/(-2)) + 1
  dist.df$scaledDc <- dist.df$scaledD
  dist.df$netcombo <- paste(substr(dist.df$V1,1,2), substr(dist.df$V2,1,2),sep = ".")
  switchlist <- c("DA.VI", "DA.SM")
  dist.df$scaledDc[dist.df$netcombo %in% switchlist] <- dist.df$scaledD[dist.df$netcombo %in% switchlist]

  plt <- plot(g, edge.width=E(g)$logp*5,
    edge.curved=dist.df$scaledDc,
    edge.color = E(g)$posneg,
    vertex.size = 3.5,
    vertex.label.family="sans",
    vertex.label.font=1,
    vertex.label.cex = 0.5,
    vertex.label = Yeo7_80verts$wnetname,
    vertex.label.dist=0.5,
    vertex.label.degree=lab.locs,
    vertex.label.color = V(g)$color)
  return(plt)
}

```

Load the ivertex and tvertex Zs for all subjects

```

## read in one exemplar to get some parameters we want
meants1 <- read_subject_meants(qced_sublists$subid[1], pint_dir, 'ivertex')
## sort the meants into the order from the new naming scheme
names(meants1) <- paste0("TR_",1:ncol(meants1))
Yeo_meants <- cbind(Yeo7_80verts,meants1)

## correlate and graph from the first one

```

```

cormat <- cor(t(select(Yeo_meants,starts_with("TR_"))))
rownames(cormat) <- Yeo_meants$SHORTNAME
colnames(cormat) <- Yeo_meants$SHORTNAME
g<-graph_from_adjacency_matrix(cormat,mode="upper",
                              weighted=T, diag=F,
                              add.rownames = "code")
g.df <- as.data.frame(get.edgelist(g), names=T)

# take the egde list as a vector
thecorrs <- E(g)$weight

# apply the Z transform (so we can do stats)
theseZ <- atanh(thecorrs)

## get two variables of interest.. edgenames and the number of edges
myedgenames <- paste(g.df[,1],g.df[,2],sep=".") ## the V1.V2 name
numedges <- length(myedgenames) ## the number of edges

## get all the Z's from ivertex
theZs_ivertex <- load_all_subject_matrices(as.character(qced_sublists$subid),
                                          myedgenames, pint_dir, 'ivertex')

## merge all those correlations back with the demographics
alldemZs_i_m <- merge(pheno,theZs_ivertex,by="subid") %>%
  gather_("Edge", "FC", myedgenames)

## get all the Z's from tvertex
theZs_tvertex <- load_all_subject_matrices(as.character(qced_sublists$subid),
                                          myedgenames, pint_dir,
                                          'tvertex')

## merge all those correlations back with the demographics
alldemZs_t_m <- merge(pheno,theZs_tvertex,by="subid") %>%
  gather_("Edge", "FC", myedgenames)

```

Run the linear model for all Template and Personalized FC edges

Note: one trick here (but justified by Saba's thesis and Graces work, is to add a DX*Sex interaction term ot all models)

```

FC_lmres <- bind_rows("Template" = alldemZs_t_m,
                     "Personalized" = alldemZs_i_m,
                     .id = "vertex_type") %>%
  group_by(Edge, vertex_type) %>%
  do(tidy(lm(FC ~ DX*Sex + mean_fd + age + Edu + Site,.))) %>%
  ungroup() %>% group_by(term) %>%
  mutate(p.FDR = p.adjust(p.value, method = "fdr")) %>%
  ungroup

```

looking at all FDR corrected results..

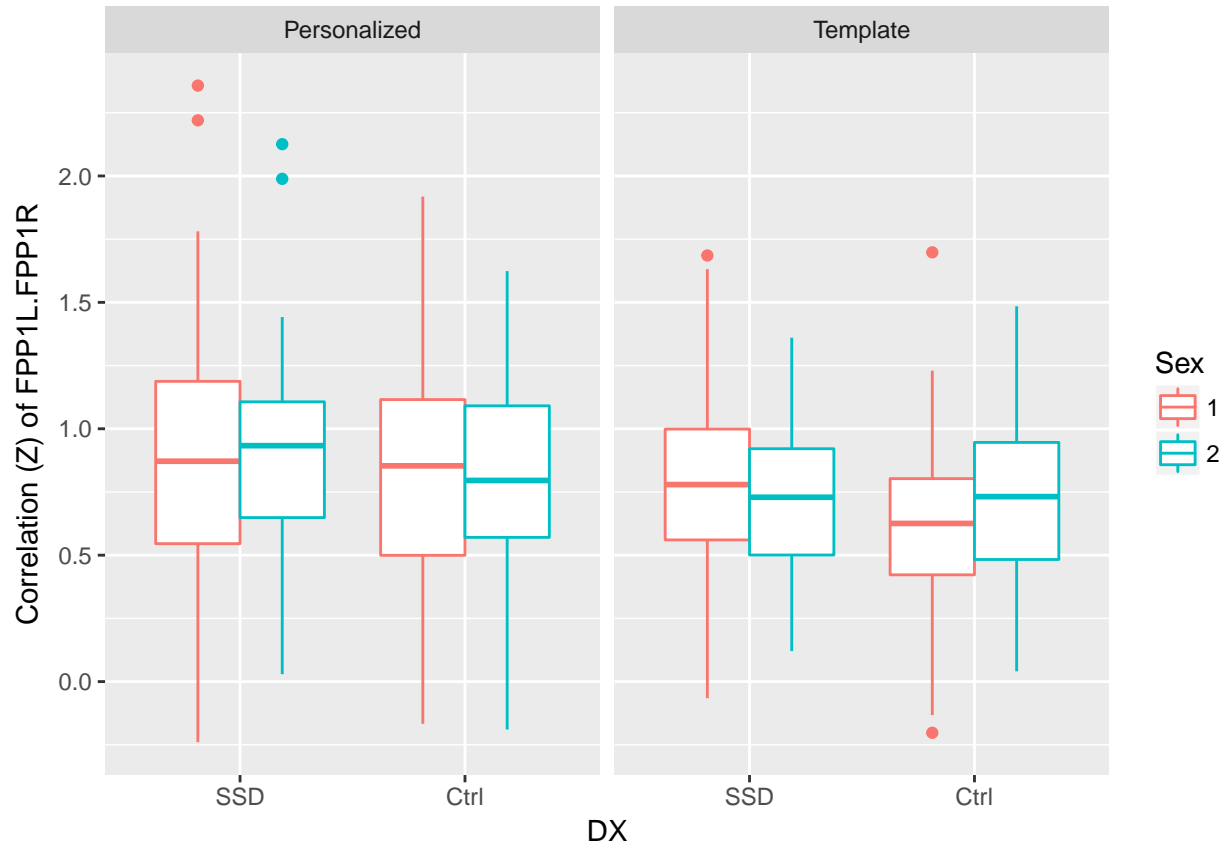
Most results are for the linear effect of age..but there is one significant effect of Diagnosis in the FP network (contralateral parietal areas..)

```
FC_lmres %>%
  filter(term %in% c("DXCtrl", "Sex2", "DXCtrl:Sex2", "age")) %>%
  arrange(p.value) %>%
  select(Edge, vertex_type, term, statistic, p.value, p.FDR) %>%
  filter(p.value < 0.0001) %>%
  kable()
```

Edge	vertex_type	term	statistic	p.value	p.FDR
DAP2R.DMP2L	Personalized	age	-4.911880	1.30e-06	0.0081221
VI04L.VI04R	Template	age	-4.514794	8.20e-06	0.0259097
FPP1L.FPP1R	Template	DXCtrl	-4.514494	8.20e-06	0.0518894
DAP2L.SMF2L	Personalized	age	4.279925	2.31e-05	0.0439448
DAP1L.VAP2R	Template	age	4.236437	2.78e-05	0.0439448
DMP1R.DMT1R	Template	Sex2	4.111470	4.71e-05	0.2486655
DAP2R.VAI1R	Personalized	age	4.027573	6.66e-05	0.0721185
DAP2L.VAI1R	Template	age	4.008427	7.21e-05	0.0721185
DMP1L.VAF3R	Personalized	Sex2	3.986851	7.87e-05	0.2486655
DAP2R.DMP2R	Personalized	age	-3.941623	9.45e-05	0.0721185

Looking at the significant effect in FP network

```
bind_rows("Template" = alldemZs_t_m,
          "Personalized" = alldemZs_i_m, .id = "vertex_type") %>%
  filter(Edge == "FPP1L.FPP1R") %>%
  ggplot(aes(x = DX, y = FC, by = Sex, color = Sex)) +
  geom_boxplot() +
  facet_grid(~vertex_type) +
  labs(y = "Correlation (Z) of FPP1L.FPP1R")
```



all the model terms within this edge

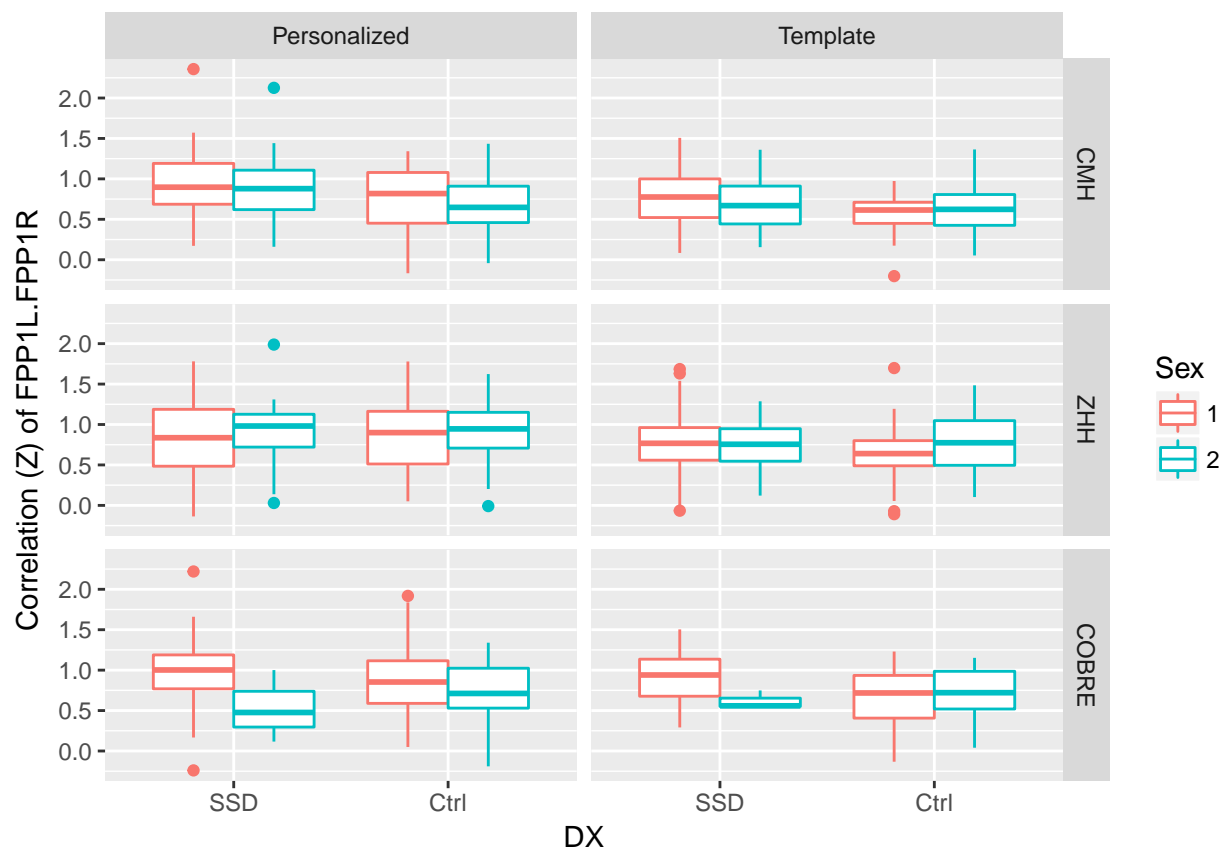
```
FC_lmres %>%
filter(Edge == "FPP1L.FPP1R") %>%
select(Edge, vertex_type, term, statistic, p.value, p.FDR) %>%
kable()
```

Edge	vertex_type	term	statistic	p.value	p.FDR
FPP1L.FPP1R	Personalized	(Intercept)	4.9027915	0.0000013	0.0000456
FPP1L.FPP1R	Personalized	DXCtrl	-0.5152849	0.6066194	0.9261453
FPP1L.FPP1R	Personalized	Sex2	-0.0991067	0.9210998	0.9981405
FPP1L.FPP1R	Personalized	mean_fd	0.6063966	0.5445722	0.7798235
FPP1L.FPP1R	Personalized	age	1.3504356	0.1775882	0.7334033
FPP1L.FPP1R	Personalized	Edu	-0.4112220	0.6811151	0.9970032
FPP1L.FPP1R	Personalized	SiteZHH	1.6322088	0.1033693	0.3993238
FPP1L.FPP1R	Personalized	SiteCOBRE	0.3332637	0.7390981	0.9129909
FPP1L.FPP1R	Personalized	DXCtrl:Sex2	-0.1388769	0.8896126	0.9977319
FPP1L.FPP1R	Template	(Intercept)	4.9213744	0.0000012	0.0000424
FPP1L.FPP1R	Template	DXCtrl	-4.5144944	0.0000082	0.0518894
FPP1L.FPP1R	Template	Sex2	-1.3698671	0.1714449	0.8081463
FPP1L.FPP1R	Template	mean_fd	-0.7201338	0.4718348	0.7280264
FPP1L.FPP1R	Template	age	-0.4186586	0.6756752	0.9377793
FPP1L.FPP1R	Template	Edu	1.7905168	0.0740758	0.9970032
FPP1L.FPP1R	Template	SiteZHH	1.1552492	0.2486314	0.5937746
FPP1L.FPP1R	Template	SiteCOBRE	2.4661057	0.0140492	0.1587799

Edge	vertex_type	term	statistic	p.value	p.FDR
FPP1L.FPP1R	Template	DXCtrl:Sex2	1.9149833	0.0561593	0.8467977

splitting the plot by Site

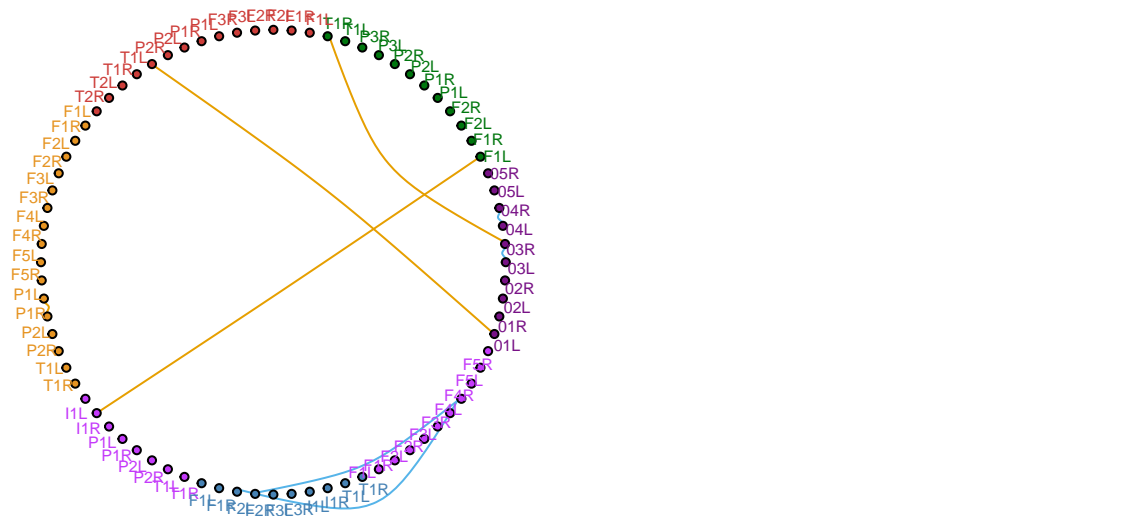
```
bind_rows("Template" = alldemZs_t_m,
          "Personalized" = alldemZs_i_m, .id = "vertex_type") %>%
  filter(Edge == "FPP1L.FPP1R") %>%
  ggplot(aes(x = DX, y = FC, by = Sex, color = Sex)) +
  geom_boxplot() +
  facet_grid(Site~vertex_type) +
  labs(y = "Correlation (Z) of FPP1L.FPP1R")
```



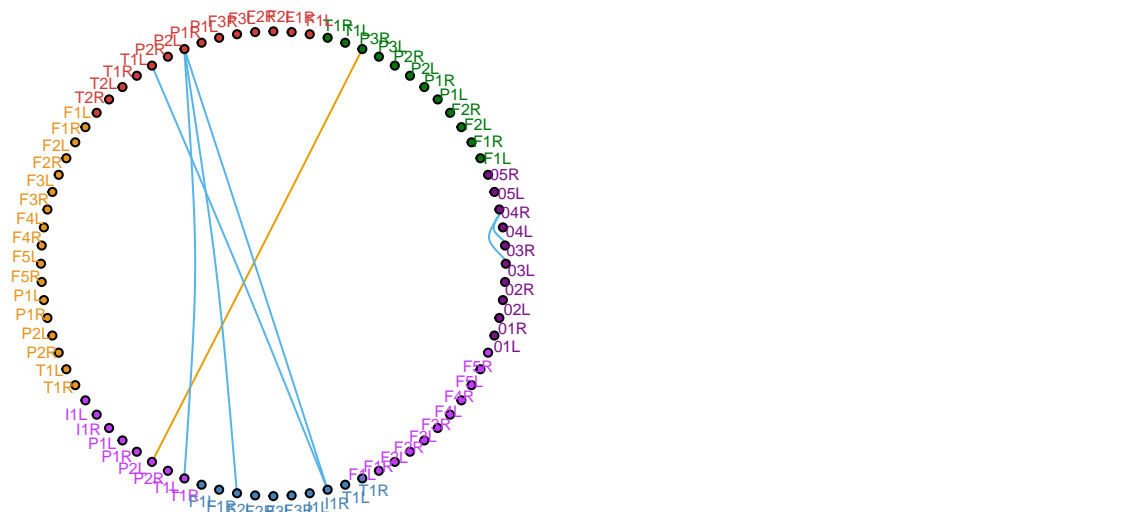
building circle plots of the subthreshold effects

```
#' threshold and draw circle shot
thresholded_circle_plot <- function(tidylm, p_thres, filter_term, filter_vtype) {
  forgraph <- tidylm %>%
    filter(term == filter_term,
           vertex_type == filter_vtype) %>%
    separate(Edge, into = c("V1", "V2"), sep = "\\\\.") %>%
    select(V1, V2, statistic, p.value) %>%
```

```
thresholded_circle_plot(FC_lmres, p_thres = 0.001,
                        filter_term = "DXCtrl", filter_vtype = "Template")
```

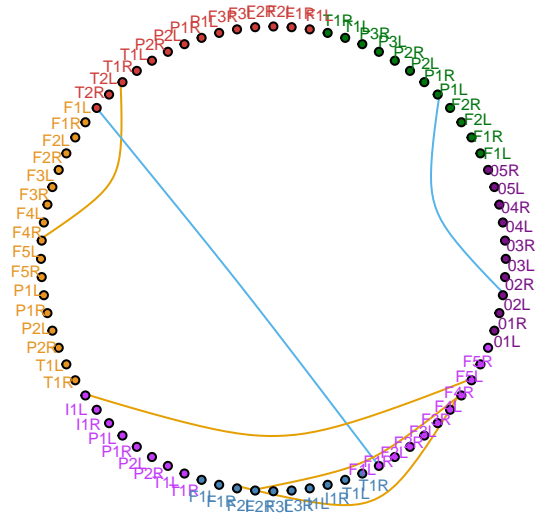


```
thresholded_circle_plot(FC_lmres, p_thres = 0.001,
                        filter_term = "DXCtrl", filter_vtype = "Personalized")
```



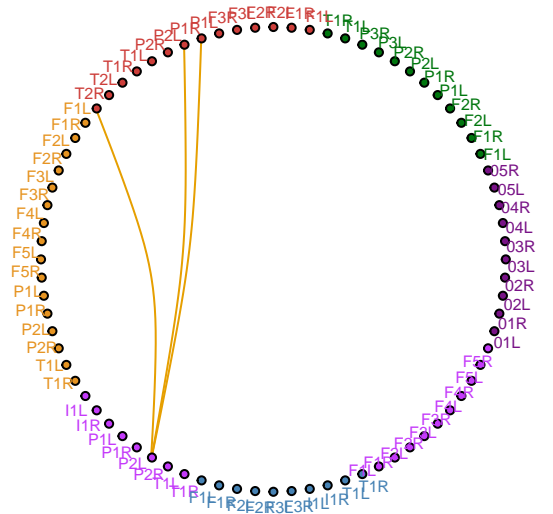
8


```
thresholded_circle_plot(FC_lmres, p_thres = 0.001,
                        filter_term = "DXCtrl:Sex2", filter_vtype = "Template")
```



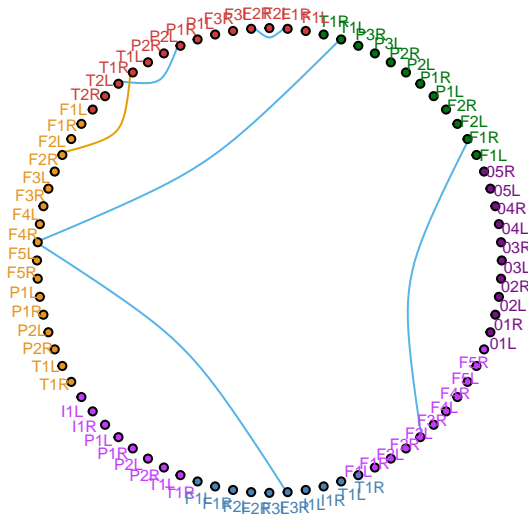
NULL

```
thresholded_circle_plot(FC_lmres, p_thres = 0.001,
                        filter_term = "DXCtrl:Sex2", filter_vtype = "Personalized")
```



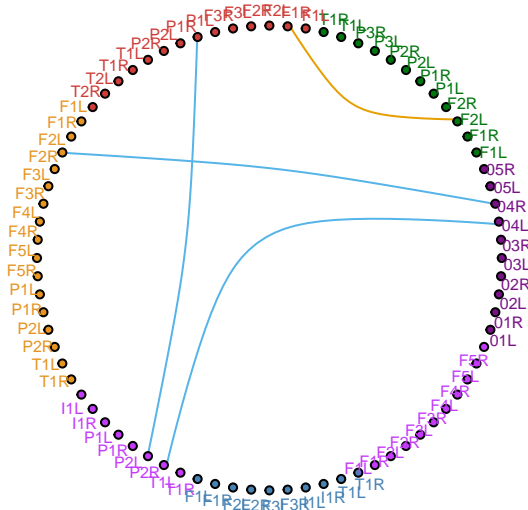
NULL

```
thresholded_circle_plot(FC_lmres, p_thres = 0.001,
                        filter_term = "Sex2", filter_vtype = "Template")
```



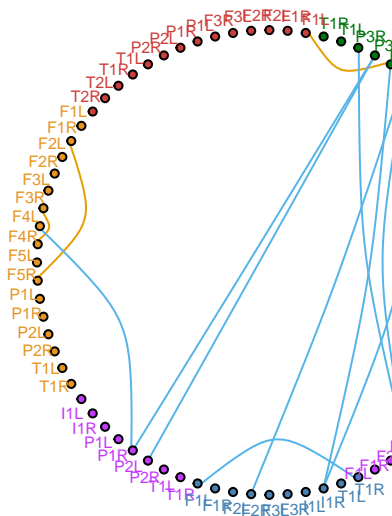
```
## NULL
```

```
thresholded_circle_plot(FC_lmres, p_thres = 0.001,
                        filter_term = "Sex2", filter_vtype = "Personalized")
```



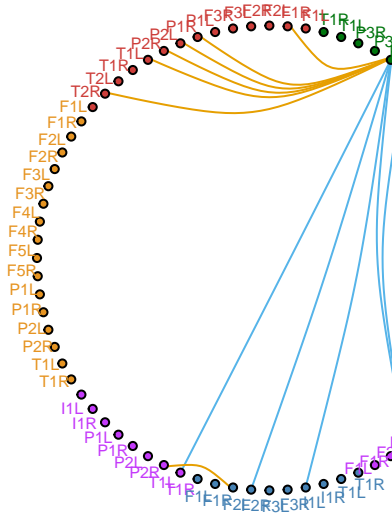
```
## NULL
```

```
thresholded_circle_plot(FC_lmres, p_thres = 0.001,
                        filter_term = "age", filter_vtype = "Template")
```



```
## NULL
```

```
thresholded_circle_plot(FC_lmres, p_thres = 0.001,
                        filter_term = "age", filter_vtype = "Personalized")
```



```
## NULL
```

New stuff for graph theory - 30 July 2017

```
# New stuff for graph theory - 30 July 2017
```

```
## merge all those correlations back with the demographics
alldemZs_i_m <- merge(qced_sublists,theZs_ivertex,by="subid")
alldemZs_t_m <- merge(qced_sublists,theZs_tvertex,by="subid")
```

```
# Averaging z-scores for each diagnostic group for the i vertices
aggregate_all_i <- aggregate(alldemZs_i_m[, 14:3173], list(alldemZs_i_m$DX_GROUP), mean)
```

```
# Subsetting into patients and controls so I can make separate adjacency matrices
```

```

aggregate_all_i_SCZ <- aggregate_all_i[1,2:3161]
aggregate_all_i_HC <- aggregate_all_i[2,2:3161]

# Averaging z-scores for each diagnostic group for the t vertices
aggregate_all_t <- aggregate(alldemZs_t_m[, 14:3173], list(alldemZs_t_m$DX_GROUP), mean)

# Subsetting into patients and controls so I can make separate adjacency matrices
aggregate_all_t_SCZ <- aggregate_all_t[1,2:3161]
aggregate_all_t_HC <- aggregate_all_t[2,2:3161]

## Thresholding the values here
# aggregate_all_i_SCZ[aggregate_all_i_SCZ < 0.60] <- 0

# Creating an iGraph - will change weights to represent different diagnostic groups
g<-graph_from_adjacency_matrix(cormat,mode="upper",
                             weighted=T, diag=F,
                             add.rownames = "code")

# Color coding the networks
V(g)$color <- ifelse(substr(V(g)$name, 0,2) == "DM", "#CD3E3A", (ifelse(substr(V(g)$name, 0,2) == "EX",

# Creating labels for the vertices - just the lobe and the hemisphere as the network is color-coded
V(g)$label_name <- substr(V(g)$name, 3,5)

# Creating adjacency matrices for i vertices
aggregate_all_i_SCZ_data <- as.matrix(aggregate_all_i_SCZ)
aggregate_all_i_SCZ_data <- as.vector(aggregate_all_i_SCZ_data)
E(g)$weight <- aggregate_all_i_SCZ_data
SCZ_i_vertex <- get.adjacency(g, type="both", attr="weight", sparse=FALSE)

aggregate_all_i_HC_data <- as.matrix(aggregate_all_i_HC)
aggregate_all_i_HC_data <- as.vector(aggregate_all_i_HC_data)
E(g)$weight <- aggregate_all_i_HC_data
HC_i_vertex <- get.adjacency(g, type="both", attr="weight", sparse=FALSE)

# Creating adjacency matrices for t vertices
aggregate_all_t_SCZ_data <- as.matrix(aggregate_all_t_SCZ)
aggregate_all_t_SCZ_data <- as.vector(aggregate_all_t_SCZ_data)
E(g)$weight <- aggregate_all_t_SCZ_data
SCZ_t_vertex <- get.adjacency(g, type="both", attr="weight", sparse=FALSE)

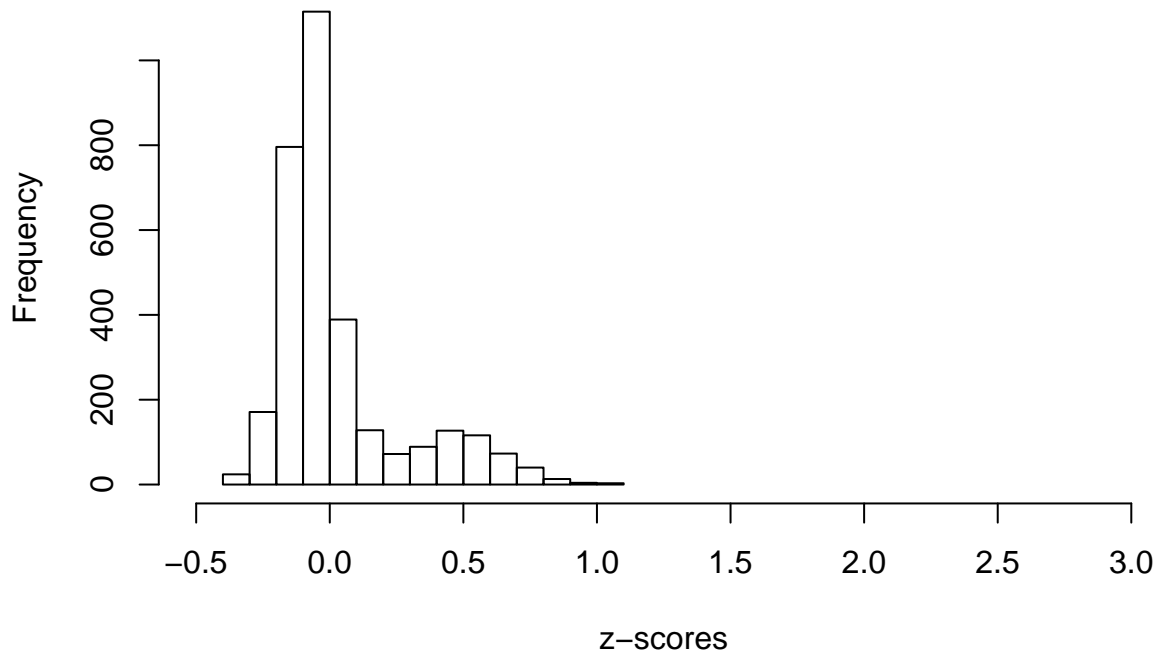
aggregate_all_t_HC_data <- as.matrix(aggregate_all_t_HC)
aggregate_all_t_HC_data <- as.vector(aggregate_all_t_HC_data)
E(g)$weight <- aggregate_all_t_HC_data
HC_t_vertex <- get.adjacency(g, type="both", attr="weight", sparse=FALSE)

# # Writing the adjacency matrices to use in MATLAB
# write.csv(SCZ_i_vertex, "../projects/saba/HCP_fMRI_SCZ/new_analyses_20170412/FC_analyses/Graph_Theory/HC")
# write.csv(HC_i_vertex, "/projects/saba/HCP_fMRI_SCZ/new_analyses_20170412/FC_analyses/Graph_Theory/HC")
# write.csv(SCZ_t_vertex, "/projects/saba/HCP_fMRI_SCZ/new_analyses_20170412/FC_analyses/Graph_Theory/SCZ")
# write.csv(HC_t_vertex, "/projects/saba/HCP_fMRI_SCZ/new_analyses_20170412/FC_analyses/Graph_Theory/HC")

```

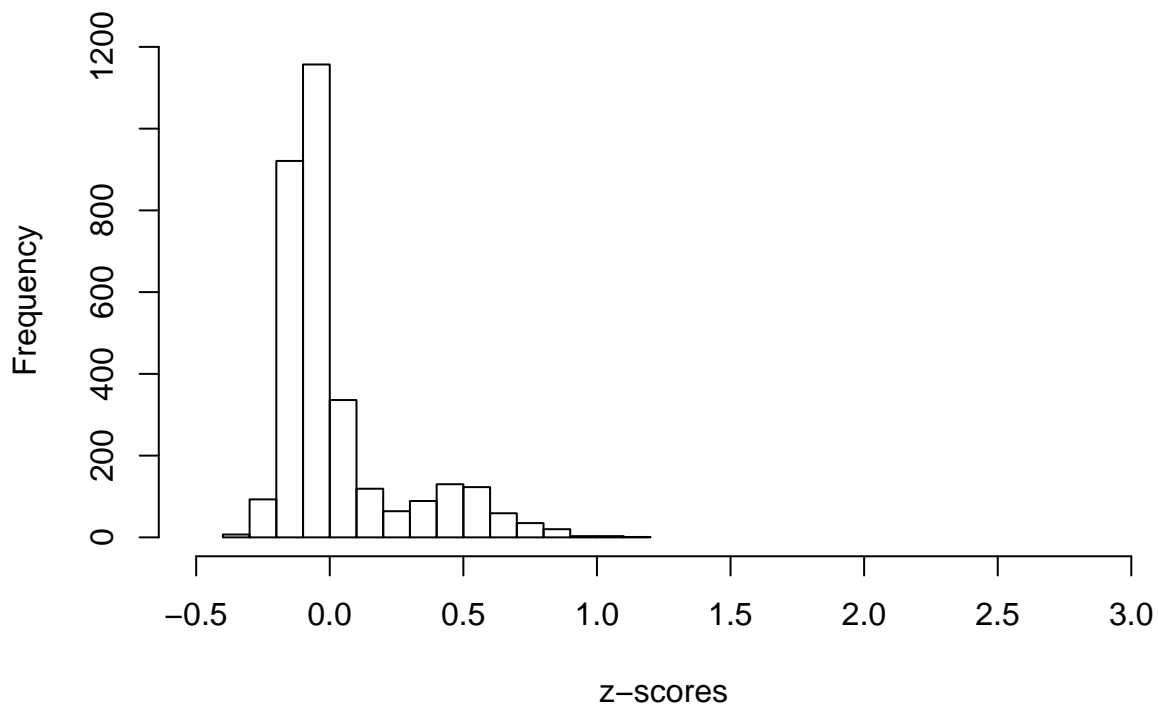
```
#plot.igraph(g,vertex.label=V(g)$name,layout=layout.fruchterman.reingold, edge.color="black",edge.width
hist(aggregate_all_i_SCZ_data, main="SCZ: i-vertex", xlab = "z-scores", xlim=c(-0.5, 3.0))
```

SCZ: i-vertex



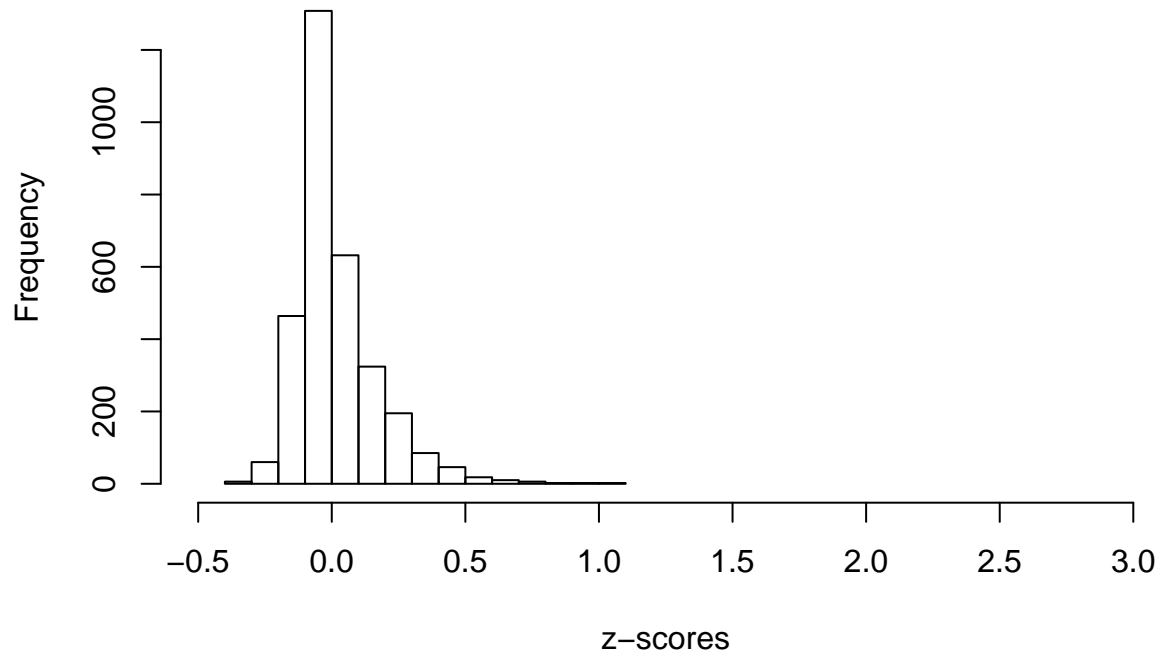
```
hist(aggregate_all_i_HC_data, main="HC: i-vertex", xlab = "z-scores", xlim=c(-0.5, 3.0))
```

HC: i-vertex



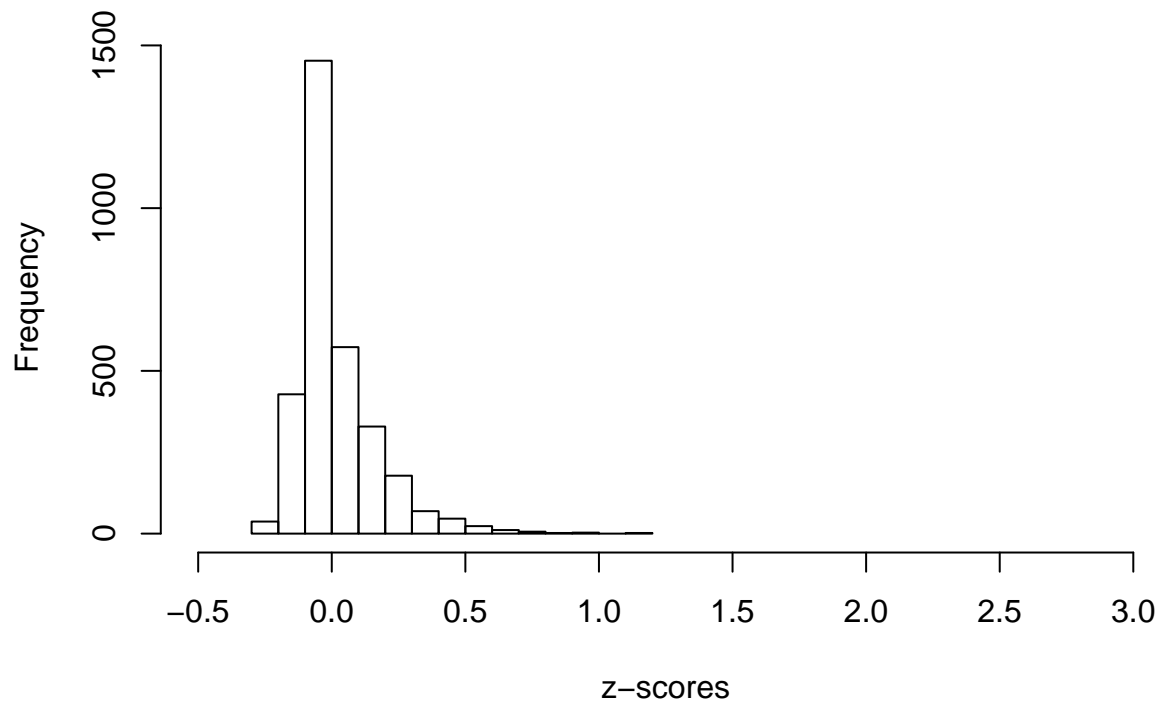
```
hist(aggregate_all_t_SCZ_data, main="SCZ: t-vertex", xlab = "z-scores", xlim=c(-0.5, 3.0))
```

SCZ: t-vertex



```
hist(aggregate_all_t_HC_data, main="HC: t-vertex", xlab = "z-scores", xlim=c(-0.5, 3.0))
```

HC: t-vertex




```
##      8      8      9      9      5      2      11      11      10      10      7      5
## DMT2L DMT2R FPF1L FPF1R FPF2L FPF2R FPF3L FPF3R FPF4L FPF4R FPF5L FPF5R
##      9      8      8      8      6      6      6      8      10      6      12      10
## FPP1L FPP1R FPP2L FPP2R FPT1L FPT1R VAF1L VAF1R VAF2L VAF2R VAF3L VAF3R
##      3      5      11      9      5      3      14      17      9      11      5      8
## VAF4L VAF4R VAF5L VAF5R VAI1L VAI1R VAP1L VAP1R VAP2L VAP2R VAT1L VAT1R
##     14     15     11      7     19     21      8     10     14     12      3      3
## SMF1L SMF1R SMF2L SMF2R SMF3L SMF3R SMI1L SMI1R SMT1L SMT1R VIO1L VIO1R
##      7      5      6      7      6      9     10     10      7      9      7      7
## VIO2L VIO2R VIO3L VIO3R VIO4L VIO4R VIO5L VIO5R
##      9      9      7      7      9      9      9      9
```

```
betweenness(g_few_vert)
```

```
## VAF1R VAF4R VAI1L VAI1R
##      0      0      0      0
```

```
degree(g_few_vert)
```

```
## VAF1R VAF4R VAI1L VAI1R
##      3      3      3      3
```

```
# Ratio of the number of edges and the number of possible edges
edge_density(g_few)
```

```
## [1] 0.1075949
```

```
## rm the extra object in memory
```

```
#rm(cormat, meants1, theZs_tvertex, theZs_iverter, Yeo_meants, g)
```