**Cross Ties in Structural Brain Networks and Alzheimer’s Disease**

Coco Luo Haoxin

Advisor: Dr. Kenneth Berenhaut

May 7, 2021

**Abstract**

This project analyzed the cross-hemisphere connections in structural brain networks of people with and without Alzheimer’s disease. The dataset of individual brains was obtained from Boniha et al. and their study on disruptions of brain anatomical connectivity. Providing information on 120 subjects who were all recruited from the local community, we are interested in finding brain regions that contribute most to (are impacted most by) the Alzheimer’s, and the connectome in the whole structural brain networks. We also examined if characteristics like age, gender and education might be related to people getting the disease. Given all 148 brain nodes, we first evaluated how a certain node is connected to all the other node in a person’s brain. Through image plots, we found that the cross ties tell some interesting story about the network. This assumption is verified by relationships shown in scatterplots between left and right pairs. All our following analyses were based on the scaled cross ties data restricted on the adjacency matrix. We than obtained some good PaLD graphs for the cognitive normal group and Alzheimer’s group in terms of diagnosis, education, and gender. Trying to find out the most homogeneous graphs and the exact regions contributing to the disorder, we took different sized samples from the cross ties and evaluated the assortativity score. Finally, we divided the cross ties into subsets according to different lobes to look for the function of some cross ties rather than the whole data. We found that brain regions that serve the function of memory and cognition are highlighted. Those include the temporal lobe and the Limbic system. We also find that the temporal inferior, the Orbital, the temporal pole, and the circular insula anterior are nodes contribute to the disease. Not surprisingly, all those regions are related to cognition, memory processing, and decision making. Further research would need to be done before we find out the exact nodes or regions of the brain that explain the onset of the disorder the most.

**Introduction**

Alzheimer's is the most common cause of dementia, a general term for memory loss and other cognitive abilities serious enough to interfere with daily life. Alzheimer's disease accounts for 60-80% of dementia cases. The greatest known risk factor us increasing age, and the majority of people with the disorder are 65 or older. Alzheimer’s worsens over time. In the early stages, memory loss is mild. But with late-stage Alzheimer’s, individuals lose the ability to carry on conversation respond to their environment. Alzheimer’s changes typically begin in the part of the brain that affects learning. As Alzheimer's advances through the brain, it leads to increasingly severe symptoms, including disorientation, mood and behavior changes; deepening confusion about events, time and place; unfounded suspicions about family, friends and professional caregivers; more serious memory loss and behavior changes; and difficulty speaking, swallowing and walking. In this study, we would look through 74 different brain regions to better figure out which areas in the brain are related to the disorder.

The data we are studying from is provided by Boniha and other contributors in 2015. it is comprised of 120 subjects recruited from a local community in South Carolina with ages between 55-90. The connectivity of 148 brain nodes across both hemispheres was recorded. The brain connectome is derived from the Diffusion tensor imaging, which may be influenced by methodological factors related to signal processing, MRI scanners and biophysical properties of neuroanatomical regions. Individuals went through 3 MRU scanning sessions within a short interval. The scanning sessions included similar T1 weights and DTI sequences. Other characteristics of the subject such as the subjects id, date of the examination, a viscode, an indicator of the severity of the disorder, age (ranging from 55 to 90), gender, education (numbers ranging from 12 to 20), ethnicity (white or Hispanic), marriage status (Married, single or Widow), CDRSB and MMSE were recorded. The variables we are interested in looking at were just the severity of the disorder (which we will refer to as diagnosis in the following paper), age, gender and education. The goal of this project is to use the Partitioned Local Depth (i.e., PaLD) to explore how brain nodes are connected in a structural sense based on the information of 120 participants. We tried different measures to split the nodes into groups of “good” and “bad”, where good represent healthy brain and “bad” suggests malfunctioning nodes that result in Alzheimer’s disease. In addition, we took different sized random samples from the across matrix (positions of same region across hemisphere weights) and implemented multiple trials to find out the highest assortative scores we could get.

**Backgrounds**

Pald was proposed by Dr. Berenhaut and Dr. Moore. Given a set of data, we would like to know how subjects are related in structural sense, or in communities. In social setting, communities arise from a balance of conflict and alignment and so we view individuals as existing at a set of distances on a social space. The local community depth refers to one’s position within a community relative to others in general. And the partitioned local depth, or simply cohesion, is the strength of alignment of a point, say w, to another point, say x, in the same set who are in conflicts. When two individuals are in conflict, the one who stands closer to one point than the other will join the conflict. The cohesion of w to x is defined to be Cx,w = P(z=w and d(Z,x) < d(Z, Y)). We can get the cohesion matrix by computing cohesion for all points. Adding the cohesions will give us the local depth. Furthermore, if w’s contribution to local depth of x is greater than expected to a typical point X from a typical focus-point, Wx, Y, we say that w to x is particularly strong. The threshold indicating particularly strong relation is half the average of the diagonal. By weighting edges according to mutual cohesion and cutting ties that are weak, the remaining components give us the information about how much the communities fit together and a perspective of what means to be a cluster.

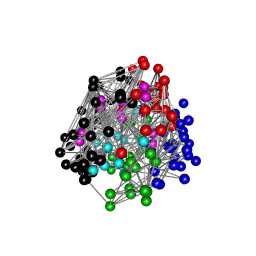
Concepts about network structure properties that maybe related to Alzheimer’s are analyzed. These could include local properties or global properties. The network we are interested in is made up of nodes, vertices and are connected by edges. The edges can be directed, or undirected. A weighted graph is a network of which a number (weight) is assigned to each edge. The weight could represent different information based on different context. The degree assigns a key score based on the number of links held by each node. It shows us the number of direct one hop connects each node has to other nodes in the network. We use it to find very connected individuals, those who are likely to hold most information or individuals who can quickly connect with the wider network. It is the simplest measure of node connectivity. We also consider about the betweenness, which measures the number of times a node lies on the shortest path between other nodes. The shortest path is a path between two nodes in a graph such that the sum of the weights of the constituent edges is minimized. The betweenness shows us which nodes are bridges between nodes in a network by identifying all the shortest paths and then counting how many times each node falls on one. This can be used to find the individuals who influence the flow around the system. The closeness scores each node though their closeness to all other nodes in the network, it computes the shortest paths between all nodes and then assigns each node a score dependent on its sum of shortest path. The closeness measures help us figure out the individuals who are best placed to influence the entire network most quickly. We also examined centrality measures as we want to know if there are certain nodes that have a high or low centrality for Alzheimer’s. We looked for 49 centrality measures in total.

**Analysis processes**

First things first, we used the 3d plots to construct the 3d brain image of the first individual. This gives us a sense on how the weighted edges are connection between each other and across hemispheres. We restricted our focus to only the top 5% greatest edge weights given the enormous number of subtle connections in a structural brain network. The output of my images is shown in figure 1.

**Figure 1: Top 5% edge weights**

A picture containing set, different, several, arranged

Description automatically generated 

We also used histograms to figure out the distribution of the frequency of the log edges’ weight for all the 120 subjects. Figure 2 only shows the first 20 and last 20 individuals. We can see that most of the frequencies of those individuals have similar distributions, (unimodal and slightly right skewed) Some of them, such as 5, 102, 111 – 120 have their peaks earlier than other subjects. We hope that by comparing the histogram between the normal people and those with AD, we can find something interesting, for example, the normal ones may have peaks earlier. However, it seems that there are no clear patterns exists in the normal people group and the AD group.

**Figure 2: Frequencies of the first 20 and last 20 individual log (edge weights)**

Diagram

Description automatically generated Diagram

Description automatically generated

Next, we considered different approaches to figure out the overall structural brain networks and how brain nodes are connected to each other. We first ran and stored a list of centrality measures (where we looked for the results of centrality measures that give us better score). We looked for rank, sum, shortest paths and betweenness, but eventually we chose to focus on the adjacency matrix. An adjacency matrix is a squared matrix of which each elements of the matrix indicate whether pairs of vertices are adjacent or not in the graph. All the 120 patients’ information were stored into a list of graphs. We later translated that information into a large 120 by 148\*148 weighted adjacency matrix to better figure out how individual brain regions are connected. We failed to gain any useful insights from the matrix on the original structure of the data. We first used image plots to convert each matrix into a graph. Each plot shows the connection from one place in the brain to the other. We found some interesting yellow blocks located at the top right corner. In figure 3, those regions are in the black circle. We noticed that those light areas are in the cross ties. For example, from the 15th image plot, we found that the region goes close to the same on the other side. 15 on the other side is 15+74 which is 89. Take another example, the 9th image plot shows similar patterns on the other side which is 9+74 = 83. And the other side of 12 is 86. Looking at the x-axis, we could see that the location of those yellow blocks roughly goes with the region on the cross side of the hemisphere. Those results suggested that something must be happened for the cross.

**Figure 3: image plot examples of cross tie relationships**

A picture containing chart

Description automatically generated Chart

Description automatically generated with low confidence Chart, histogram

Description automatically generated

To verify our assumptions, we further checked the scatterplot on the cross ties, which are nodes that located at similar place across hemispheres. From the scatterplot outputs, we were excited to find out that there are some strong relationships between the left and right pairs in the brain, so something interesting do happen for the cross ties. Figure 4 gave some left and right pairs are clearly connected. Those cross ties include, the occipital superior, the orbital, the precentral, the frontal inferior, the occipital middle and Lunatus, and the orbital shaped.

**Figure 4: scatter plots of strong pairs**

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Because of our previous findings revealed patterns happening for the cross ties, we decided to focus our study on the crosstie data. hoping to find out regions that contribute to the pattern and how to distinguish areas that are important in terms of Alzheimer’s disease. Two different approaches were used to scale the matrix. We first scaled our graph information with the number of standard deviations. We also used the minmax function, which only contain 0: smallest and 1: largest in the scaled matrix. We hope that the scaled data could make it easier for us to split the good and the bad individuals and get better representations of groups of individuals who are like each other. After the min max scaling, we ran PaLD on the data. Figure 5 gives us the general layout of the split of the groups in terms of the good and bad diagnosis, difference in gender and different levels of education. For “figure 1: diagnosis,” Green numbers correspond to the good (cognitive normal) and red ones correspond to the bad (Alzheimer’s Disorder). The output looks pretty good. It seems that the red ones are kind of in the middle and the green normal ones are at the peripheral. The figure shows that the 43, 23, 31, 26, 21st individuals are similar and 114, 102, 120, 115, 119 are similar. In a good (cognitive normal) way.

**Figure 5: cross tie splits in terms of diagnosis, gender, and education**

A picture containing scatter chart

Description automatically generated Diagram

Description automatically generatedDiagram

Description automatically generated

With the original graph, we further examined the what the split would be like for different genders and education group. It seems that males (green ones) have higher chance to get the disease, and people who has higher education (pink group) seems to have a higher change to get the disease than those who don’t (green group).

Finally, we wrote an assortativity function that would allow us to take different sized random samples from the scaled cross matrix and then decide on the number of times we would like to run the function. We looked for the highest assortativity score and the top contributing regions for each trial. I choose to run on samples of 7 and 5. We ran the trails repeatedly for about 300 times. The highest score I got is 0.42 and the most frequently occurred regions are: temporal inferior, which is known to be a region essential in recognizing patterns, faces, and objects; the

Orbital, known to be the region for cognitive process of decision-making; the temporal pole, an association cortex involved with multimodal analysis, especially in social and emotional processing, and circular insula anterior, which is mainly for sensory processing and sensory binding. So far we have just look for all cross ties, but what if only some cross ties are contributing? We thus determined to look at some subsets, and one natural subset would be looking for the lobe.

**Figure 6: assortativity scores for different lobes**

**Chart, diagram

Description automatically generated** **Chart

Description automatically generated** Chart, diagram

Description automatically generated

**Chart

Description automatically generated** **Chart

Description automatically generated** A picture containing sky

Description automatically generated

It seems that there are no interesting things happened for the crosstie data in terms of different lobes. The assotativity score I got for frontal lobe is the highest, about 0.216, then Limbi which is 0.147, then Parietal 0.111, Temporal 0.051, Insula 0.036, and Occipital which is roughly 0. Although we didn’t find anything significant, the idea is that looking for subsets would be a great way to find out the more detailed information on how the regions are connected.

**Discussion and Conclusion**

Our study shows that there are several strong connections happening across hemispheres which might contribute to Alzheimer’s disorder. We find that the temporal and limbic lobe contribute more to the disorder than other regions in the brain. The temporal lobe is largely responsible for creating and preserving both conscious and long-term memory. And the Limbic system, is a set of structures in the brain that deal with emotions and memory, that regulates autonomic or endocrine function in response to emotional stimuli. It is not a surprised since **Alzheimer‘s** is a progressive disease, where dementia symptoms gradually worsen over a number of years. In its early stages, memory loss is mild, but with late-stage **Alzheimer’s**, individuals lose the ability to carry on a conversation and respond to their environment. We also find a couple of specific nodes that may serve as a stronger indication of the disorder, temporal inferior, the Orbital, the temporal pole, and the circular insula anterior. All those regions are related to cognition, memory processing, and decision making. However, we still need further research to verify our results.

There were several reasons we did not get a great split. Obviously, we need more subjects for our study, we need more cognitive normal individuals and individuals with Alzheimer’s to compare with. We would like to restrict our study on a particular region in the brain rather than the whole brain. This will narrow our focus and probably find some nodes that serve significant purpose within that region. If we have enough time, we would also like to explore more options on different network properties. Rather than looking only at the adjacency matrix, we could probably explore more options in the local properties such as closeness, degree etc.

Last but not the least, we would like to explore different types of subset and try to find some interesting connections among those groups.

**Appendix A (codes):**

library("igraph")

library("RColorBrewer")

library("CINNA")

library(StatMatch)

library(igraph)

library(expm)

library(CINNA)

library(markovchain)

library(readxl)

library(quantable)

library(brainGraph)

library(rgl)

**### ----------------------------- functions ------------------------- ###**

## random graph with same number of nodes (sample from all possible edges)

randgraph<-function (n,e=n) {

g2<-random.graph.game(n,e,type="gnm")

g2<-induced.subgraph(g2,largest\_comp(g2))

g2

}

## gives us the biggest component of the graph

largest\_comp<-function(graph) {

cl <- clusters(graph)

V(graph)[which.max(cl$csize) == cl$membership]

}

## get the transition matrix

getP<-function (g) {

A<-as.matrix(get.adjacency(g))

r<-apply(A,1,sum)

P<-diag(1/r)%\*%A

P

}

## pald

pald<-function(X,dist=TRUE,scale=TRUE,show.cont=TRUE,show.plot=TRUE,bet=1,

gt=NULL,lw=1,tit=FALSE,L=NULL,lab=FALSE,...)

{

if (dist==FALSE) {

if (scale==TRUE) { X<-scale(X,scale=TRUE)[,]}

D<-dist(X) } else D<-X

D<-as.matrix(D)

getcontmat<-function(D, b=0,h=.5,bet=1,cr=1:dim(D)[1]){

D<-round(D,15)

L<-NULL

n=dim(D)[1]

A3=matrix(0,n,n)

for(x in 1:(n-1)){

for(y in (x+1):n){

dx=D[x,]; dxt=D[,x]

dy=D[y,]; dyt=D[,y]

Uxy=(which((dx<=bet\*D[x,y]) | (dy<=bet\*D[y,x]))) #the reaching set

wx<-1\*(dx[Uxy]<dy[Uxy])+h\*((dx[Uxy]==dy[Uxy]))

wy<-1\*(dy[Uxy]<dx[Uxy])+h\*((dx[Uxy]==dy[Uxy]))

A3[x,Uxy]=A3[x,Uxy]+1/(length(Uxy))\*wx

A3[y,Uxy]=A3[y,Uxy]+1/(length(Uxy))\*wy

}

}

diag(A3)<-diag(A3)+b

rownames(A3)=1:n

colnames(A3)=1:n

return(A3/(n-(b==0)))

}

if (is.null(rownames(D)[1])) {rownames(D)<-1:dim(D)[1]}

nm<-rownames(D)

B<-getcontmat(D,h=.5,b=0,bet=bet)

q<-apply(B,1,sum)

names(q)<-rownames(D)

RU<-mean(diag(B))/2

A<-B

ASym<-pmin(A, t(A))

color<-c( brewer.pal(n = 8, name = "Dark2"), brewer.pal(n=8, name="Set2"))

color<-rep("grey",16)

rownames(B)<-rownames(D);colnames(B)<-colnames(D)

diag(ASym)<-1

g<-graph.adjacency(ASym,weighted=TRUE,mode="undirected")

g<-simplify(g)

#V(g)$name<-rownames(D)

w<-E(g)$weight

E<-get.edgelist(g)

E<-E[order(w), ]

w<-w[order(w)]

g<-graph.edgelist(E, directed=FALSE)

g<-g+setdiff(as.character(1:dim(D)[1]),V(g)$name)

E(g)$weight<-w

Acut<-get.adjacency(g,attr="weight")

Acut[Acut < RU]<-0

diag(Acut)<-1

gcut<-graph.adjacency(Acut,weighted=TRUE,mode="undirected")

gcut<-simplify(gcut)

e<-as.numeric(get.edgelist(g)[, 1])

u<-igraph::clusters(gcut)$membership

u<-u[order(as.numeric(names(u)))]

edge\_colors<-color[u[e]]

edge\_colors[E(g)$weight<RU]<-"white"

if(show.cont){

edge\_colors[E(g)$weight<RU]<-"gray"}

edge\_widths<-E(g)$weight

edge\_widths[edge\_widths<RU]<-edge\_widths[edge\_widths<RU]/20

if (lab) {lab<-rownames(D)[as.numeric(V(g)$name)]} else lab<-""

V(gcut)$clusters<-igraph::clusters(gcut)$membership

if (is.null(L)) {L<-layout\_nicely(g)} else {L<-L[as.numeric(V(g)$name),]}

if(show.plot){

plot(g, ylim=c(-1, 1),xlim=c(-1, 1),...,

vertex.size=4, vertex.label.cex=1,

vertex.color=color[igraph::clusters(gcut)$membership],

# vertex.label.color=color[igraph::clusters(gcut)$membership],

vertex.label.color="black",

vertex.label=lab,

vertex.label.dist = 1, edge.width=lw\*100\*(edge\_widths),

edge.color=edge\_colors, asp=0,layout=L,

main="PaLD")

if ((!dist)&(tit))

{title(paste(abbreviate(colnames(X)),collapse=","))}

}

if(!is.null(gt[1])){

plot(g, ylim=c(-1, 1),xlim=c(-1, 1),...,

vertex.size=4, vertex.label.cex=1.2,

# vertex.color=color[igraph::clusters(gcut)$membership],

#color[igraph::clusters(gcut)$membership],

vertex.label.color=gt[as.numeric(V(g)$name)],

vertex.color=gt[as.numeric(V(g)$name)],

#vertex.label.color="black",

vertex.label=lab[as.numeric(V(g)$name)],

vertex.label.dist = 1, edge.width=lw\*100\*(edge\_widths),

edge.color=edge\_colors, asp=0,layout=L)

if ((!dist)&(tit))

{title(paste(abbreviate(colnames(X)),collapse=","))}

V(g)$gt<-gt[as.numeric(V(g)$name)]

}

V(g)$name<-rownames(D)[as.numeric(V(g)$name)]

rownames(ASym)<-rownames(B);colnames(ASym)<-colnames(B)

cl<-igraph::clusters(gcut)$membership

names(cl)<-rownames(D)[as.numeric(names(cl))]

cl<-cl[sapply(rownames(D),function(c) which(names(cl)==c))]

V(gcut)$name<-rownames(D)[as.numeric(V(gcut)$name)]

list(C=B,Cmin=ASym,g=g,g2=gcut,bound=RU,clusters=(cl),

isolated=setdiff(rownames(D),V(g)$name),layout=L,depths=(q))

}

## pald-smooth (runs pald pn N matrix, smooth the overlaid variables according to cohesion)

# averaging over the pald neighbors

paldsmooth<-function(M,V,name=rownames(M),diagz=FALSE,R=FALSE,global=TRUE,local=FALSE,original=FALSE){

M<-cbind(M)

V<-cbind(V)

Mt<-M

D<-gower.dist(Mt)

L<-pald(D,dist=TRUE,show.plot=FALSE)

if (diagz) {diag(L$C)<-0;diag(L$Cmin)<-0}

p<-apply(Mt,1,paste,collapse=",")

nm<-as.numeric(V(L$g2)$name)

if (R) dev.new()

# plot(L$g2,vertex.label.color="blue",vertex.label.cex=.8,

# layout=L$layout,vertex.size=1,vertex.label=p[nm])

R2<-NULL

hc<-colorRampPalette(brewer.pal(8, "PiYG"))(40)

V<-cbind(V)

for (i in 1:dim(V)[2])

{

vc<-V[,i]

varname<-colnames(V)[i]

col<-hc[round((vc-min(vc))/(max(vc)-min(vc))\*35)+1]

Cweightg<-diag(1/apply(L$C,1,sum))%\*%L$C

vcg<-as.vector(Cweightg%\*%vc) # PaLD smoothing

colg<-hc[round((vcg-min(vcg))/(max(vcg)-min(vcg))\*35)+1]

Clocal<-L$C

Clocal[Clocal<L$bound]<-0.001

Cweightl<-diag(1/apply(Clocal,1,sum))%\*%Clocal

vcl<-as.vector(Cweightl%\*%vc) # PaLD smoothing

coll<-hc[round((vcl-min(vcl))/(max(vcl)-min(vcl))\*35)+1]

diff<-sum( (vc-mean(vc))^2)

diffl<-sum( (vc-vcl)^2 )

diffg<-sum( (vc-vcg)^2 )

R2<-rbind(R2,c(diff,diffl,diffg,

(diff-diffl) /diff,

(diff-diffg) /diff))

titg<-c(round((diff-diffg) /diff,4))

titl<-c(round((diff-diffl) /diff,4))

if (global)

{if (R) dev.new()

plot(L$g2,vertex.label.color="black",vertex.label.cex=0.1,

layout=L$layout,vertex.size=7,vertex.color=colg[nm],vertex.label="")

title(c("global",varname,titg))}

if (local)

{if (R) dev.new()

plot(L$g2,vertex.label.color="black",vertex.label.cex=0.1,

layout=L$layout,vertex.size=7,vertex.color=coll[nm],vertex.label="")

title(c("local",varname,titl))}

if(original)

{if (R) dev.new()

plot(L$g2,vertex.label.color="black",vertex.label.cex=0.1,

layout=L$layout,vertex.size=7,vertex.color=col[nm],vertex.label="")

title(varname,titg)}

}

# plot(L$layout,type="n")

# text(L$layout,labels=name)

list(L,R2)

}

## pm produces an a x b grid of plots

pm<-function(a,b){par(mfrow=c(a,b))};

pm(1,1)

### getgraphnames ###

getgraphnames<-function(v,m=0,M=Inf) {

u<-sapply(v, function(a) {

g<-get(a);

ans<-FALSE;

if (is.igraph(g)) {

g<-simplify(getgiant(as.undirected(g)));

ans<-((vcount(g)<M)&(vcount(g)>m)&(assortativity.degree(g)>0))}

ans})

v[u][!is.na(v[u])]

v #several graphs

}

# minmax scaling

minmax <- function(v){

if(sd(v) == 0){

ans<-rep(0.5, length(v))

}

else{

ans <-(v-min(v))/(max(v)-min(v))

}

ans

}

# Define a function for plotting a matrix

myImagePlot<-

function(x, gt=1, cex.axis=1,...){

gt <- rev(gt)

min <- min(x)

max <- max(x)

yLabels <- rownames(x)

xLabels <- colnames(x)

title <-c()

# check for additional function arguments

if( length(list(...)) ){

Lst <- list(...)

if( !is.null(Lst$zlim) ){

min <- Lst$zlim[1]

max <- Lst$zlim[2]

}

if( !is.null(Lst$yLabels) ){

yLabels <- c(Lst$yLabels)

}

if( !is.null(Lst$xLabels) ){

xLabels <- c(Lst$xLabels)

}

if( !is.null(Lst$title) ){

title <- Lst$title

}

}

# check for null values

if( is.null(xLabels) ){

xLabels <- c(1:ncol(x))

}

if( is.null(yLabels) ){

yLabels <- c(1:nrow(x))

}

layout(matrix(data=c(1,2), nrow=1, ncol=2), widths=c(4,1), heights=c(1,1))

# Red and green range from 0 to 1 while Blue ranges from 1 to 0

ColorRamp <- rgb( seq(0,1,length=256), # Red

seq(0,1,length=256), # Green

seq(1,0,length=256)) # Blue

ColorLevels <- seq(min, max, length=length(ColorRamp))

# Reverse Y axis

reverse <- nrow(x) : 1

yLabels <- yLabels[reverse]

x <- x[reverse,]

# Data Map

par(mar = c(3,5,2.5,2))

image(1:length(xLabels), 1:length(yLabels), t(x), col=ColorRamp, xlab="",

ylab="", axes=FALSE, zlim=c(min,max))

if( !is.null(title) ){

title(main=title)

}

axis(BELOW<-1, at=1:length(xLabels), labels=xLabels, cex.axis=0.7)

axis(LEFT <-2, at=1:length(yLabels), labels=FALSE, las= HORIZONTAL<-1,

cex.axis=cex.axis)

Map(axis, side=2, at=1:length(yLabels), col.axis=gt, labels=yLabels, lwd=0, las=1,cex.axis=cex.axis)

# Color Scale

par(mar = c(3,2.5,2.5,2))

image(1, ColorLevels,

matrix(data=ColorLevels, ncol=length(ColorLevels),nrow=1),

col=ColorRamp,

xlab="",ylab="",

xaxt="n")

layout(1)

}

# getcolor does a color ramp for v (from dark purple[low] to dark green[high])

# plot(1:10, col=getcolour(1:10), pch=17, cex=2)

getcolour<-function(v){

hc<-colorRampPalette(brewer.pal(8, "PiYG"))(35)

vc<-as.numeric(v)

col<-hc[round((vc-min(vc))/(max(vc)-min(vc))\*35)+1]

col}

Rcolors<-c(12,17,24,26,30,41,50,55,72,73,78,

81,98,99,107,117,125,132,139,145,150,257,374,381,403,430,

450,451,456,461,465,469,477,491,500,504,516,548,

554,564,578,586,596,613,619,632,638,642,657)

pm(1,1)

plot(cbind(0,1:length(Rcolors)),type="n")

text(cbind(0,1:length(Rcolors)),labels=colors()[Rcolors])

text(cbind(0,1:length(Rcolors)),labels=colors()[Rcolors],col=colors()[Rcolors])

# color options we can use

colourv<-c("green","blue","purple","orange","red")

**### ------------------------ centrality computation -------------------- ###**

# g1 can be used to see all the centralities

g1<- randgraph(10)

# some of the centrality measures are excluded

exc<-proper\_centralities(g1)[c(6,31,33,38,43)]

cent\_compute<-function(g1,prcent=setdiff(proper\_centralities(g1),exc)){

# CINNA centrality measures

print(g1$name)

M2<-calculate\_centralities(g1, include = prcent)

inc2<-which(as.vector(unlist(lapply(M2,length)))==vcount(g1))

M<-as.data.frame(M2[inc2])

colnames(M)<-names(M2[inc2])

M<-data.frame(M)

P<-getP(g1)

MC <- new("markovchain",transitionMatrix = P,name = "MC")

D <- meanFirstPassageTime(MC)

Pald\_short<-pald(shortest.paths(g1),dist=TRUE,show.plot=FALSE)$depths # Pald - shortest path

Pald\_hitTo<-pald(D,dist=TRUE,show.plot=FALSE)$depths # Pald - hitting time to

Pald\_hitFrom<-pald(t(D),dist=TRUE,show.plot=FALSE)$depths # Pald - hitting time from

M$Pald\_short = Pald\_short

M$Pald\_hitTo = Pald\_hitTo

M$Pald\_hitFrom = Pald\_hitFrom

# rownames(M) <- paste(c(rownames(M),g1$name),collapse=".")

M <- as.matrix(M)

M

}

**#### --------------------- read in the DataTSClean file ---------------------- ####**

data("destrieux")

# dest is all the information about the brain regions (locations)

dest<-as.data.frame(destrieux)

rownames(dest)<-1:dim(dest)[1]

coords3d<-as.matrix(destrieux[,2:4]) #3d coordinates

### ----- read in data --------###

setwd("~/Desktop/PALD")

# L2data is the patient information

L2data<-read.csv(file="DataTSClean2.csv",header=TRUE,row.names=1)

**## ---------------- Reading in each individual network (L2) --------------- ##**

setwd("~/Desktop/PALD/BL\_NET")

v<-list.files()

### L2 is a 120 long list of graphs with patients' information (age, weights, ...)

L2 <-list()

ML2 <- NULL

for (i in 1:length(v)){

A<-as.matrix(read\_excel(v[i],col\_names=FALSE))

A<-apply(A,2,as.numeric)

rownames(A)<-1:dim(A)[1]

colnames(A)<-1:dim(A)[1]

ML2<-rbind(ML2,as.vector(A))

g<-graph.adjacency(A,weighted=TRUE)

g<-as.undirected(g)

g$name<-v[i]

g$A <- A

g$dg<-as.character(L2data$DX\_bl[i])

g$age<-as.numeric(L2data$AGE[i])

g$gender<-as.character(L2data$PTGENDER[i])

g$education<-as.character(L2data$PTEDUCAT[i])

q<-as.numeric(L2data$DX\_bl[i])

g$dgn<-q

if(q==1){g$dgn<-5}

if(q==2){g$dgn<-1}

if(q==3){g$dgn<-3}

if(q==4){g$dgn<-4}

if(q==5){g$dgn<-2}

V(g)$name<-sapply(V(g)$name,function(a) gsub("V","",a))

g$name <- i

L2[[i]] <- g

}

# check to assure everything is working

i<-sample(120,1);g<-L2[[i]];cbind(t(L2data[i,c(1,4:7)]),c(g$name,g$dg,g$age,g$gender,g$education));print(g$dgn)

## M is the matrix, its rows represent different graphs

M<-t(sapply(L2,function(g) apply(g$A,1,rank))) #this is the weight matrix

M<-t(sapply(L2,function(g) apply(g$A,1,sum))) #this is the weight matrix

M<-t(sapply(L2,function(g) as.vector(shortest.paths(g))))

M <-t(sapply(L2,function(g) as.vector(betweenness(g))))

### this is what we focused on: 120 by 21904(148\*148)

M <-t(sapply(L2,function(g) as.vector(get.adjacency(g,attr="weight"))))

#### vectors of patient's characteristics

# 1: cog normal 2: SMC 3: EMCI 4: LMCI 5: AD

dgn <- sapply(L2,function(g) g$dgn)

dg <- sapply(L2,function(g) g$dg)

# range 55 - 90.3

age <- sapply(L2,function(g) g$age)

# males = 1, females = 2

gender <- (sapply(L2,function(g) g$gender)=="Male")+1

# range 11 to 20

education <- sapply(L2,function(g) as.numeric(g$education))

good<- (dgn==1)|(dgn==5) # subset of individuals with particular characteristics

#good <- good & (gender == 1)

#good<-(dgn==1)

#bad<-(dgn==5)

#compare<-(dgn==1)|(dgn==5)

Mg<-M[good,] ## Mg is the graph data only for particular patients(good)

#Mg<-M

Mts<-t(scale(t(Mg))) #Mts is the scaled graph information with the # of sd

Mts <- t(apply(Mg,1,minmax)) #Mts is the scaled graph information with minmax (smalles:0, biggest:1)

# for 148 x 148 data, vacross is the positions of the same region cross hemisphere weights

vacross<-148\*(0:73)+75+(0:73)

# random pairing as compared to the original

assmat<-NULL

for (i in 1:400){

#vacross <- 148\*(0:73)+75+sample(0:73)

vacross<-148\*(0:73)+75+(0:73)

vacross.s<-sample((74),9)

# Macross is Mts restricted to only cross ties (74 columns)

#vacross<-assmat[300,-1][1:3]

Macross<-Mts[,vacross[vacross.s]]

Macross<-t(apply(Macross, 1, minmax))

# dendrogram for Macross data

rownames(Macross) <- which(good)

#hcl <- hclust(dist(Macross), method = "complete")

#hcl\_ord<-hcl$order

#plot(hcl)

#myImagePlot(Macross[hcl\_ord,],col.lab="green")

#myImagePlot(Macross[order(dgn[good]),],gt=sort(dgn[good]))

## L\_dgn is a pald list for Macross data

#L\_dgn<-pald(cbind(Macross),dist=FALSE,scale=FALSE, show.plot=FALSE,gt=colourv[dgn[as.numeric(rownames(Macross))]], main="diagnosis", show.cont=FALSE,lab=TRUE)

L\_dgn<-pald(cbind(Macross),dist=FALSE,scale=FALSE, show.plot=FALSE, main="diagnosis", show.cont=FALSE,lab=TRUE)

# g1 is the cohesion graph, g2 is threshold

g1<-L\_dgn$g;g2<-L\_dgn$g2

A<-get.adjacency(g1,attr="weight")

v<-as.numeric(V(g1)$name)

A<-A[order(v),order(v)]

g1<-graph.adjacency(A,weighted=TRUE,mode="undirected")

A<-get.adjacency(g2,attr="weight")

v<-as.numeric(V(g2)$name)

A<-A[order(v),order(v)]

g2<-graph.adjacency(A,weighted=TRUE,mode="undirected")

lay<-L\_dgn$layout

lay<-lay[order(v),]

#plot(g2,vertex.label.color=colourv[dgn[good]],layout=lay,vertex.size=.8

# ,vertex.label.cex=1.5,vertex.label=which(good))

ass<-assortativity(g2, dgn[good])

assmat<-rbind(assmat,c(ass,vacross.s))

title(round(ass,2))

print(c(i,ass))

}

assmat<-assmat[order(assmat[,1]),];tail(assmat)

n<-dim(assmat)[1]

vacross.s<-assmat[n,-1]

vacross.s<-1:74

Macross<-Mts[,vacross[vacross.s]]

Macross<-t(apply(Macross, 1, minmax))

# dendrogram for Macross data

rownames(Macross) <- which(good)

#hcl <- hclust(dist(Macross), method = "complete")

#hcl\_ord<-hcl$order

#plot(hcl)

#myImagePlot(Macross[hcl\_ord,],col.lab="green")

myImagePlot(Macross[order(dgn[good]),],gt=sort(dgn[good]))

## L\_dgn is a pald list for Macross data

#L\_dgn<-pald(cbind(Macross),dist=FALSE,scale=FALSE, show.plot=FALSE,gt=colourv[dgn[as.numeric(rownames(Macross))]], main="diagnosis", show.cont=FALSE,lab=TRUE)

L\_dgn<-pald(cbind(Macross),dist=FALSE,scale=FALSE, show.plot=FALSE, main="diagnosis", show.cont=FALSE,lab=TRUE)

# g1 is the cohesion graph, g2 is threshold

g1<-L\_dgn$g;g2<-L\_dgn$g2

A<-get.adjacency(g1,attr="weight")

v<-as.numeric(V(g1)$name)

A<-A[order(v),order(v)]

g1<-graph.adjacency(A,weighted=TRUE,mode="undirected")

A<-get.adjacency(g2,attr="weight")

v<-as.numeric(V(g2)$name)

A<-A[order(v),order(v)]

g2<-graph.adjacency(A,weighted=TRUE,mode="undirected")

lay<-L\_dgn$layout

lay<-lay[order(v),]

plot(g2,vertex.label.color=colourv[dgn[good]],layout=lay,vertex.size=.8

,vertex.label.cex=1.5,vertex.label=which(good))

ass<-assortativity(g2, dgn[good])

#assmat<-rbind(assmat,c(ass,vacross.s))

destrieux[vacross.s,]

####################################################

rownames(Mts) <- which(good)

runn<-function(n,z)

{

assvec<-NULL

for (i in 1:300){

vacross <- 148\*(0:73)+75+sample(0:73)

vacross <- 148\*(0:73)+75+(0:73)%%74

z<-25

vacross<-sample(vacross, z)

#vacross<-assvec[300,3:(2+z)]

# Macross is Mts restricted to only cross ties (74 columns)

Macross<-Mts[,vacross]

#Macross<-Mts

Macross<-scale(Macross)

Macross<-t(apply(Macross, 1, minmax))

colnames(Macross)<-vacross

# dendrogram for Macross data

rownames(Macross) <- which(good)

hcl <- hclust(dist(Macross), method = "complete")

hcl\_ord<-hcl$order

#plot(hcl)

a<-dgn[which(good)]

#dev.new();myImagePlot(Macross[order(dgn[good]),],gt=rev(sort(a)))

#dev.new();myImagePlot(Macross[hcl\_ord,],gt=rev(a[hcl\_ord]))

## L\_dgn is a pald list for Macross data

L\_dgn<-pald(cbind(Macross),dist=FALSE,scale=FALSE, show.plot=FALSE,gt=colourv[dgn[good]],

show.cont=FALSE,lab=TRUE)

# g1 is the cohesion graph, g2 is threshold

g1<-L\_dgn$g;g2<-L\_dgn$g2

A<-get.adjacency(g1,attr="weight")

v<-as.numeric(V(g1)$name)

A<-A[order(v),order(v)]

g1<-graph.adjacency(A,weighted=TRUE,mode="undirected")

A<-get.adjacency(g2,attr="weight")

v<-as.numeric(V(g2)$name)

A<-A[order(v),order(v)]

g2<-graph.adjacency(A,weighted=TRUE,mode="undirected")

lay<-L\_dgn$layout

lay<-lay[order(v),]

#plot(g2,vertex.label.color=colourv[dgn[good]],layout=lay,vertex.size=.8

# ,vertex.label.cex=1.5,vertex.label=which(good))

#title("Figure 1: diagnosis")

assort<-assortativity(g2, dgn[good])

title(assort)

destname<-dest$name[(vacross-74)%%148]

assvec<-rbind(assvec, c(assort,mean(Macross),sort(vacross),sort((vacross-74)%%148)))

print(round(sort(assvec[,1]),2))

}

assvec<-assvec[order(assvec[,1]),]

#list(assvec=assvec, vacross=vacross,destname=destname)

#}

assvec[order(assvec[,1]),]

# ------- image plots for different groups

## compare different groups in education

vector\_group1 <- c(77,62,35,45,109,103,65)

myImagePlot(Macross[vector\_group1,]); title(paste(vector\_group1, collapse = ", "))

vector\_group2 <- c(77,62,35,45,109,103,65)

myImagePlot(Macross[vector\_group2,]); title(paste(vector\_group2, collapse = ", "))

## compare different groups in diagnosis

dng\_group1 <- c(62,35,45,109,55,103,65) #bad 1

myImagePlot(Macross[dng\_group1,]); title(paste(c(dng\_group1, "bad"), collapse = ", "))

dng\_group4 <- c(62,35,45,109,55,103,65, 1, 54, 89, 2, 34) #bad 2

myImagePlot(Macross[dng\_group4,]); title(paste(c(dng\_group4, "bad"), collapse = ", "))

dng\_group2 <- c(114, 102, 119, 120, 115) #good

myImagePlot(Macross[dng\_group2,]); title(paste(c(dng\_group2,"good"), collapse = ", "))

dng\_group3 <- c(43, 28, 21, 31, 26) #good group2

myImagePlot(Macross[dng\_group3,]); title(paste(c(dng\_group3,"good"), collapse = ", "))

dng\_group5<- which(bad)

myImagePlot(Macross[dng\_group5,]); title(paste(c(dng\_group5, "bad"), collapse = ", "))

dng\_group6<- which(good)

myImagePlot(Macross[dng\_group6,]); title(paste(c(dng\_group6, "bad"), collapse = ", "))

dng\_group7<- c(90, 64, 89, 1, 54, 2, 34, 32, 35, 45, 109, 103, 65, 55, 14, 29, 3, 118, 117, 19, 100, 22, 10, 15, 24)

myImagePlot(Macross[dng\_group7,]); title(paste(c(dng\_group5, "bad"), collapse = ", "))

Appendix B:

Table1: diagnosis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AD** | **CN** | **EMCI** | **LMCI** | **SMC** |
| 25 | 23 | 38 | 20 | 14 |

Table 2: diagnosis vs. gender

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **gender** | **AD** | **CN** | **EMCI** | **LMCI** | **SMC** |
| Male | 11 | 11 | 12 | 9 | 9 |
| Female | 14 | 12 | 26 | 11 | 5 |

Table 3: diagnosis vs. education

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EDU** | **AD** | **CN** | **EMCI** | **LMCI** | **SMC** |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 5 | 2 | 4 | 4 | 0 |
| 13 | 1 | 1 | 3 | 0 | 0 |
| 14 | 1 | 3 | 4 | 0 | 2 |
| 15 | 2 | 0 | 2 | 1 | 1 |
| 16 | 6 | 5 | 7 | 8 | 2 |
| 17 | 1 | 1 | 3 | 0 | 0 |
| 18 | 4 | 6 | 7 | 2 | 3 |
| 19 | 1 | 3 | 2 | 3 | 3 |
| 20 | 3 | 2 | 5 | 1 | 3 |

**49 Centrality measures**

[1] "subgraph centrality scores"

[2] "Topological Coefficient"

[3] "Average Distance"

[4] "Barycenter Centrality"

[5] "BottleNeck Centrality"

[6] "Centroid value"

[7] "Closeness Centrality (Freeman)"

[8] "ClusterRank"

…

[42] "Load Centrality"

[43] "Flow Betweenness Centrality"

[44] "Information Centrality"

[45] "Dangalchev Closeness Centrality"

[46] "Group Centrality"

[47] "Harmonic Centrality"

[48] "Local Bridging Centrality"

[49] "Wiener Index Centrality"

**References:**

Andrew Disney, 2nd J. 2020. (2021, February 8). *Social network analysis: Centrality measures*. Cambridge Intelligence. https://cambridge-intelligence.com/keylines-faqs-social-network-analysis/.

Berenhaut, K.S., Moore, K.E. and Melvin, R.L., A social perspective on perceived distances

reveals deep community structure, 2020. in revision.

Berenhaut, K. S., Barr, P. S., Kogel, A. M., & Melvin, R. L. (2018). Cluster-based network proximities for arbitrary nodal subsets. *Scientific Reports*, *8*(1). <https://doi.org/10.1038/s41598-018-32172-0>

Blondel, V. D., Gajardo, A., Heymans, M., Senellart, P., & Van Dooren, P. (2004). A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching. *SIAM Review*, *46*(4), 647–666. https://doi.org/10.1137/s0036144502415960

Bonilha L, Gleichgerrcht E, Fridriksson J, Rorden C, Breedlove JL, Nesland T, et al. (2015) Reproducibility of the Structural Brain Connectome Derived from Diffusion Tensor Imaging. PLoS ONE 10(9): e0135247. https://doi.org/10.1371/journal.pone.0135247

Leicht, E. A., Holme, P., & Newman, M. E. (2006). Vertex similarity in networks. *Physical Review E*, *73*(2). https://doi.org/10.1103/physreve.73.02612