Investigating Potential Pathways in Disease Progression of Alzheimer's Disease

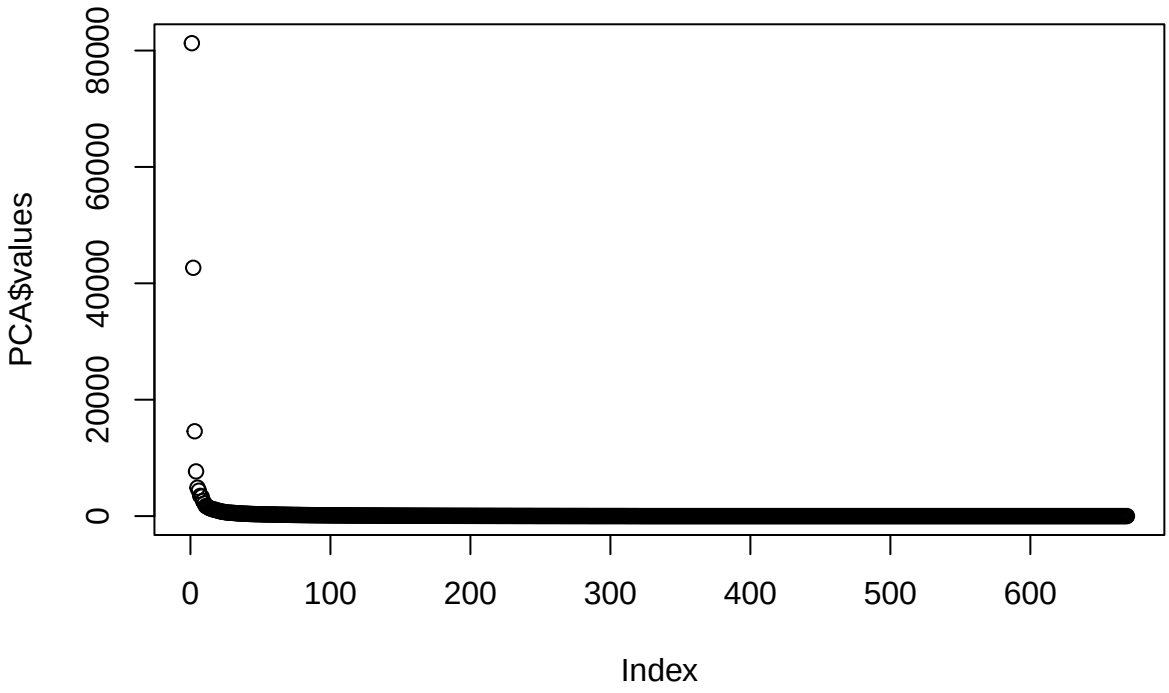
Zhentao Yu

2022-12-5

Read data:

Extract common factors:

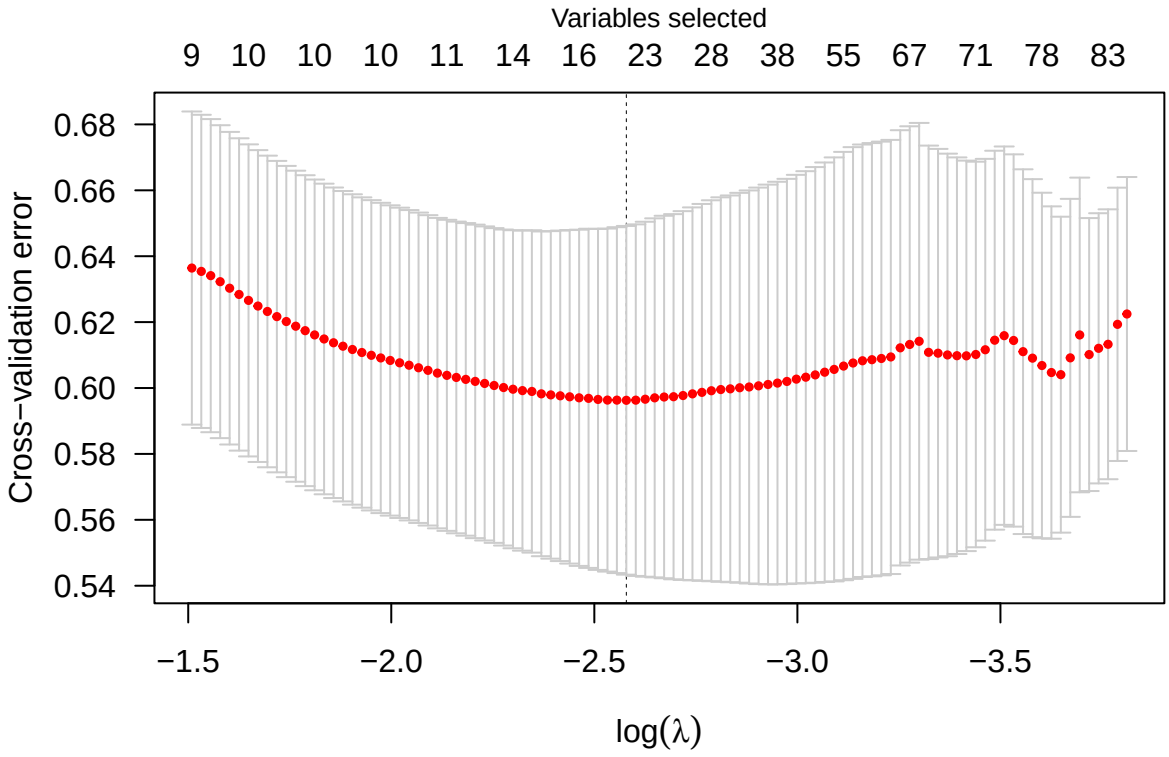
```
PCA   <- eigen(X %*% t(X))
K     <- choose.K(X,PCA) # 1st way to choose K: 9
plot(PCA$values)
```



Estimation of F, L, U and process of cross validation:

```
F.hat <- sqrt(n) * PCA$vectors[, 1:K]
L.hat <- t(X) %*% F.hat / n
U.hat <- X - F.hat %*% t(L.hat)
```

```
pen <- c(rep(1, ncol(U.hat)), rep(0, ncol(F.hat))) # penalty factor
cv <- cv.ncvreg(cbind(U.hat, F.hat), Y, family = 'gaussian', seed=1,lambda.min=0.1,nfold=5,penalty = 'S
                penalty.factor = pen)
plot(cv)
```




```
cv$lambda.min#0.0758
```

```
## [1] 0.07586428
```

With the tuned $\lambda = 0.07586428$ from previous section, we refit the model with SCAD penalty and select the variables with non-zero estimates as the final set of predictors.

```
fit <- ncvreg(cbind(U.hat, F.hat), Y, family = 'gaussian', lambda=cv$lambda.min, penalty = 'SCAD',
              penalty.factor = pen,
              standardize = F)

beta.hat <- head(as.vector(fit$beta), - K)[-1]
blank1<-cbind(blank1,sum(beta.hat!=0))
blank2<-cbind(blank2,which(beta.hat!=0))
idx.nonzero <- which(fit$beta[-1]!= 0)
idx.nonzero <- idx.nonzero[1:(length(idx.nonzero)-K)]
nonzeros <- as.matrix(fit$beta[idx.nonzero+1], ncol = 1)
nonzeros <- cbind(nonzeros, paste('Node.', idx.nonzero, sep = ''))
rownames(nonzeros) <- colnames(X)[idx.nonzero]
```

```
save(dat, Y, adj, X, K, fit, nonzeros, file = 'MEM_output.RData')
print(ascii(nonzeros, include.rownames = TRUE, include.colnames = FALSE,
            format = "f", digits = 3,
            caption = ""), type = "org")
```

```
## #+CAPTION:
## | Amyloid.Left_G_temporal_inf        | -0.0547937022712578  | Node.37  |
## | Amyloid.Right_G_pariet_inf.Angular | 0.0046126010737875   | Node.99  |
## | Amyloid.Left.Putamen               | -0.0881697059222472  | Node.151 |
## | Amyloid.Right.Putamen              | -0.0106796163011445  | Node.157 |
## | Amyloid.Right.Pallidum             | -0.0476484576943478  | Node.158 |
## | FDG.Left_G_cingul.Post.dorsal      | 0.304747419386679    | Node.169 |
## | FDG.Left_S_front_inf               | 0.00794163906273882  | Node.212 |
## | FDG.Right_G_oc.temp_med.Lingual    | -0.00955349943823486 | Node.256 |
## | FDG.Right_G_parietal_sup           | -0.0262090556855152  | Node.261 |
## | FDG.Right_S_subparietal            | 0.000826098880021223 | Node.305 |
## | FDG.Left.Hippocampus               | 0.0255279582110518   | Node.313 |
```

Mediation Analysis:

Then we find potential mediators in each pathway:

```
out1 <- matrix(NA, 5, 6)
for (i in 3:7) {
  for (j in 1:6) {
    mod <- paste("\nY~b *", colnames(X_adj2)[7 + j], '+ c *', colnames(X_adj2)[i], '+ ', colnames(X_adj
                 colnames(X_adj2)[7 + j], '~a *', colnames(X_adj2)[i], '\n',
                 'direct := c\n',
                 'indirect := a * b\n',
                 "total := c + (a * b)\n ")

    fit <- sem(model = mod, data = dat1)
    out1[i-2, j] <- summary(fit)$PE$pvalue[length(summary(fit)$PE$pvalue)-1]
  }
}
rownames(out1) <- colnames(X_adj2)[3:7]
colnames(out1) <- colnames(X_adj2)[8:13]

indirect <- matrix(NA, 5, 6)
for (i in 3:7) {
  for (j in 1:6) {
    mod <- paste("\nY~b *", colnames(X_adj2)[7 + j], '+ c *', colnames(X_adj2)[i], '+ ', colnames(X_adj
                 colnames(X_adj2)[7 + j], '~a *', colnames(X_adj2)[i], '\n',
                 'direct := c\n',
                 'indirect := a * b\n',
                 "total := c + (a * b)\n ")

    fit <- sem(model = mod, data = dat1)
    indirect[i-2, j] <- summary(fit)$PE$est[15]/summary(fit)$PE$se[15]
  }
}
rownames(indirect) <- colnames(X_adj2)[3:7]
colnames(indirect) <- colnames(X_adj2)[8:13]
```

```
indirectp <- matrix(NA, 5, 6)
for (i in 3:7) {
  for (j in 1:6) {
    mod <- paste("\nY~b *", colnames(X_adj2)[7 + j], '+ c *', colnames(X_adj2)[i], '+ ', colnames(X_adj
                 colnames(X_adj2)[7 + j], '~a *', colnames(X_adj2)[i], '\n',
                 'direct := c\n',
                 'indirect := a * b\n',
                 "total := c + (a * b)\n ")

    fit <- sem(model = mod, data = dat1)
    indirectp[i-2, j] <- summary(fit)$PE$pvalue[15]
  }
}
rownames(indirectp) <- colnames(X_adj2)[3:7]
colnames(indirectp) <- colnames(X_adj2)[8:13]
adjusted_idp<-matrix(NA,5,6)
for (i in 1:5){
  adjusted_idp[i,]<-qvalue(indirectp[i,],fdr.level = 0.05,pi0=1)$qvalues
}
rownames(adjusted_idp)<-rownames(indirectp)
colnames(adjusted_idp)<-colnames(indirectp)
```

```
print(ascii(out1, include.rownames = TRUE, include.colnames = TRUE,
            format = "f", digits = 3,
            caption = ""), type = "org")
```

```
## #+CAPTION:
## |                                  | FDG.Left_G_cingul.Post.dorsal | FDG.Left_S_front_inf | FDG.Rig
## | Amyloid.Left_G_temporal_inf      | 0.000                         | 0.016                | 0.139
## | Amyloid.Right_G_pariet_inf.Angular | 0.000                       | 0.076                | 0.134
## | Amyloid.Left.Putamen             | 0.606                         | 0.245                | 0.865
## | Amyloid.Right.Putamen            | 0.707                         | 0.453                | 0.987
## | Amyloid.Right.Pallidum           | 0.001                         | 0.066                | 0.545
```

```
out.D <- ifelse(out1 > 0.05, 'D', 'M')

print(ascii(out.D, include.rownames = TRUE, include.colnames = TRUE,
            format = "f", digits = 3,
            caption = ""), type = "org")
```

```
## #+CAPTION:
## |                                  | FDG.Left_G_cingul.Post.dorsal | FDG.Left_S_front_inf | FDG.Rig
## | Amyloid.Left_G_temporal_inf      | M                             | M                    | D
## | Amyloid.Right_G_pariet_inf.Angular | M                           | D                    | D
## | Amyloid.Left.Putamen             | D                             | D                    | D
## | Amyloid.Right.Putamen            | D                             | D                    | D
## | Amyloid.Right.Pallidum           | M                             | D                    | D
```

```
adjusted1<-matrix(NA,5,6)
for (i in 1:5){
  adjusted1[i,]<-qvalue(out1[i,],fdr.level = 0.05,pi0=1)$qvalues
```

```
}
rownames(adjusted1)<-rownames(out.D)
colnames(adjusted1)<-colnames(out.D)
out.q <- ifelse(adjusted1 > 0.05, 'D', 'M')
out.D<-as.data.frame(out.D)
```