Real Data Analysis Final

5/30/2022

Contents


```
## load the pseudo-data and the 'grivet' package
load("pseudo_gene_expr_data.RData")
library(grivet)
```

```
## This function judeges if the given `U` encodes a directed acyclic graph.
## Arguments:
## U: the causal effect matrix.
## Returns:
## flag: the encoded graph has a cycle if flag = 0, acyclic if flag = 1.
is.acyclic <- function(U){
  flag <- 1
  while (sum(U)>0){
    if (min(colSums(U))>0){
```

```
      flag <- 0
      break
    }
    idx <- which(colSums(U)!=0)
    U <- U[idx,idx,drop=FALSE]
  }
  return(flag)
}
```

AD part

```
## centralize the data for AD
Y1 <- Y.AD
X1 <- X.AD
Y1 <- t(t(Y1)-colMeans(Y1))
X1 <- t(t(X1)-colMeans(X1))
```

Estimation of V matrix

```
set.seed(0)
## tuning parameters for V matrix estimation
tau.list <- c(0.01,0.02,0.03)
gamma.list <- seq(0.00001,0.001,0.00001)
n.fold <- 5
result1.1 <- cv.intdag.pmle.diff.aic(X1,Y1,tau.list,gamma.list,n.fold) ## estimate V
```

Structure Recovery

```
V <- result1.1$V ## the estimated V
result1.2 <- topological_order(V) ## recover the DAG structure
```

Coefficient estimation

```
Pi1 <- result1.2$an_mat ## The matrix representing ancestral relationship, 1 if existing, 0 otherwise.
Phi1 <- result1.2$in_mat ## The matrix representing invertion relationship, 1 if existing, 0 otherwise.
Piv1 <- result1.2$iv_mat ## The matrix representing candidate IV set, 1 if existing, 0 otherwise.
```

```
set.seed(0)
## tuning parameters for parameter estimation
n.fold <- 5
tau.list <- c(0.01,0.02,0.03)
gamma.list <- seq(0.1,3.5,0.1)
result1.3 <- cv.intdag.coe(X1,Y1,Pi1,Phi1,Piv1,tau.list,gamma.list,n.fold) ## estimate the parameters
```

mbtlp for precision matrix estimation

```
set.seed(0)
Z1 <- Y1 - Y1%*%result1.3$U - X1%*%result1.3$W ## compute the residuals
## tuning parameters for estimating the support of precision matrix
tau.list <- c(0.01,0.02,0.03)
gamma.list <- seq(0,0.0001,0.000001)
```

```
n.fold <- 5
result1.5 <- cv.MB_Union(Z1,tau.list,gamma.list,n.fold) ## estimate the support of precision matrix
```

```
S <- result1.5$S ## support of the precision matrix
Sigma <- result1.3$Sigma ## covariance matrix of the residuals
max.it <- 10000 ## the number of maximum iterations for BCD algorithm
tol <- 1e-7 ## the stopping criterion for BCD algorithm
wi1 <- precision_refit(Sigma,S,max.it,tol) ## the estimated precision matrix
```

APP -> APOE

```
## indexes for test genes
idx1 <- which(colnames(Y1) == 'APP')
idx2 <- which(colnames(Y1) == 'APOE')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi1),ncol(Pi1))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi1+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat1.1 <- intdag.2lr(X1,Y1,U0,Phi1,U1,Phi1,wi1)$statistic
stat1.1
```

```
## [1] 1625.844
```

LRP1 -> CASP3

```
## indexes for test genes
idx1 <- which(colnames(Y1) == 'LRP1')
idx2 <- which(colnames(Y1) == 'CASP3')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi1),ncol(Pi1))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi1+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat1.2 <- intdag.2lr(X1,Y1,U0,Phi1,U1,Phi1,wi1)$statistic
stat1.2
```

```
## [1] 353.0208
```

APP -> APBB1

```
## indexes for test genes
idx1 <- which(colnames(Y1) == 'APP')
idx2 <- which(colnames(Y1) == 'APBB1')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi1),ncol(Pi1))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi1+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat1.3 <- intdag.2lr(X1,Y1,U0,Phi1,U1,Phi1,wi1)$statistic
stat1.3
```

```
## [1] 26.56038
```

CAPN1 -> CDK5R1

```
## indexes for test genes
idx1 <- which(colnames(Y1) == 'CAPN1')
idx2 <- which(colnames(Y1) == 'CDK5R1')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi1),ncol(Pi1))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi1+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat1.4 <- intdag.2lr(X1,Y1,U0,Phi1,U1,Phi1,wi1)$statistic
stat1.4
```

```
## [1] 406.627
```

LRP1 -> GSK3B

```
## indexes for test genes
idx1 <- which(colnames(Y1) == 'LRP1')
idx2 <- which(colnames(Y1) == 'GSK3B')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi1),ncol(Pi1))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi1+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat1.5 <- intdag.2lr(X1,Y1,U0,Phi1,U1,Phi1,wi1)$statistic
stat1.5
```

```
## [1] 0.06711445
```

CAPN1 -> CASP3

```
## indexes for test genes
idx1 <- which(colnames(Y1) == 'CAPN1')
idx2 <- which(colnames(Y1) == 'CASP3')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi1),ncol(Pi1))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi1+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat1.6 <- intdag.2lr(X1,Y1,U0,Phi1,U1,Phi1,wi1)$statistic
stat1.6
```

```
## [1] 1.659737
```

ATP5F1 -> CASP3

```
## indexes for test genes
idx1 <- which(colnames(Y1) == 'ATP5F1')
idx2 <- which(colnames(Y1) == 'CASP3')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi1),ncol(Pi1))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi1+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat1.7 <- intdag.2lr(X1,Y1,U0,Phi1,U1,Phi1,wi1)$statistic
stat1.7
```

```
## [1] 0.7055343
```

CN Part

```
## centralize the data for CN
Y2 <- Y.CN
X2 <- X.CN
Y2 <- t(t(Y2)-colMeans(Y2))
X2 <- t(t(X2)-colMeans(X2))
```

Estimation of V matrix

```
set.seed(0)
## tuning parameters for V matrix estimation
tau.list <- c(0.01,0.02,0.03)
gamma.list <- seq(0.00001,0.001,0.00001)
n.fold <- 5
result2.1 <- cv.intdag.pmle.diff.aic(X2,Y2,tau.list,gamma.list,n.fold) ## estimate V
```

Structure Recovery

```
V <- result2.1$V ## the estimated V
result2.2 <- topological_order(V) ## recover the DAG structure
```

Coefficient estimation

```
Pi2 <- result2.2$an_mat ## The matrix representing ancestral relationship, 1 if existing, 0 otherwise.
Phi2 <- result2.2$in_mat ## The matrix representing invertion relationship, 1 if existing, 0 otherwise.
Piv2 <- result2.2$iv_mat ## The matrix representing candidate IV set, 1 if existing, 0 otherwise.
```

```
set.seed(0)
## tuning parameters for parameter estimation
n.fold <- 5
tau.list <- c(0.01,0.02,0.03)
gamma.list <- seq(0.1,3.5,0.1)
result2.3 <- cv.intdag.coe(X2,Y2,Pi2,Phi2,Piv2,tau.list,gamma.list,n.fold) ## estimate the parameters
```

mbtlp for precision matrix estimation

```
set.seed(0)
Z2 <- Y2 - Y2%*%result2.3$U - X2%*%result2.3$W ## compute the residuals
## tuning parameters for estimating the support of precision matrix
tau.list <- c(0.01,0.02,0.03)
gamma.list <- seq(0,0.0001,0.000001)
n.fold <- 5
result2.5 <- cv.MB_Union(Z2,tau.list,gamma.list,n.fold) ## estimate the support of precision matrix
```

```
S <- result2.5$S ## support of the precision matrix
Sigma <- result2.3$Sigma ## covariance matrix of the residuals
max.it <- 10000 ## the number of maximum iterations for BCD algorithm
tol <- 1e-7 ## the stopping criterion for BCD algorithm
wi2 <- precision_refit(Sigma,S,max.it,tol) ## the estimated precision matrix
```

APP -> APOE

```
## indexes for test genes
idx1 <- which(colnames(Y2) == 'APP')
idx2 <- which(colnames(Y2) == 'APOE')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi2),ncol(Pi2))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi2+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat2.1 <- intdag.2lr(X2,Y2,U0,Phi2,U1,Phi2,wi2)$statistic
stat2.1
```

```
## [1] 0.2002287
```

LRP1 -> CASP3

```
## indexes for test genes
idx1 <- which(colnames(Y2) == 'LRP1')
idx2 <- which(colnames(Y2) == 'CASP3')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi2),ncol(Pi2))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi2+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat2.2 <- intdag.2lr(X2,Y2,U0,Phi2,U1,Phi2,wi2)$statistic
stat2.2
```

```
## [1] 0.1203953
```

APP -> APBB1

```
## indexes for test genes
idx1 <- which(colnames(Y2) == 'APP')
idx2 <- which(colnames(Y2) == 'APBB1')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi2),ncol(Pi2))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi2+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat2.3 <- intdag.2lr(X2,Y2,U0,Phi2,U1,Phi2,wi2)$statistic
stat2.3
```

```
## [1] 1144.698
```

CAPN1 -> CDK5R1

```
## indexes for test genes
idx1 <- which(colnames(Y2) == 'CAPN1')
idx2 <- which(colnames(Y2) == 'CDK5R1')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi2),ncol(Pi2))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi2+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat2.4 <- intdag.2lr(X2,Y2,U0,Phi2,U1,Phi2,wi2)$statistic
stat2.4
```

```
## [1] 96.31005
```

LRP1 -> GSK3B

```
## indexes for test genes
idx1 <- which(colnames(Y2) == 'LRP1')
idx2 <- which(colnames(Y2) == 'GSK3B')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi2),ncol(Pi2))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi2+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat2.5 <- intdag.2lr(X2,Y2,U0,Phi2,U1,Phi2,wi2)$statistic
stat2.5
```

```
## [1] 0.6024297
```

CAPN1 -> CASP3

```
## indexes for test genes
idx1 <- which(colnames(Y2) == 'CAPN1')
idx2 <- which(colnames(Y2) == 'CASP3')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi2),ncol(Pi2))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi2+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat2.6 <- intdag.2lr(X2,Y2,U0,Phi2,U1,Phi2,wi2)$statistic
stat2.6
```

```
## [1] 181.1635
```

ATP5F1 -> CASP3

```
## indexes for test genes
idx1 <- which(colnames(Y2) == 'ATP5F1')
idx2 <- which(colnames(Y2) == 'CASP3')
```

```
## structure under the alternative hypothesis
U.test <- matrix(0,nrow(Pi2),ncol(Pi2))
U.test[idx1,idx2] <- 1
U1 <- 1*((Pi2+U.test)>0)
is.acyclic(U1)==0 ## judge if the acyclicity constraint is violated
```

```
## [1] FALSE
```

```
## structure under the null hypothesis
U0 <- U1-U.test
```

```
## compute the test statistic for the hypothesis
stat2.7 <- intdag.2lr(X2,Y2,U0,Phi2,U1,Phi2,wi2)$statistic
stat2.7
```

```
## [1] 147.6889
```

Summary of Results

```
## summarize the results
stat1 <- c(stat1.1,stat1.2,stat1.3,stat1.4,stat1.5,stat1.6,stat1.7)
stat2 <- c(stat2.1,stat2.2,stat2.3,stat2.4,stat2.5,stat2.6,stat2.7)
stat.mat <- cbind(stat1,stat2)
colnames(stat.mat) <- c("AD","CN")
rownames(stat.mat) <- c("APP -> APOE","LRP1 -> CASP3","APP -> APBB1","CAPN1 -> CDK5R1","LRP1 -> GSK3B",'
```

```
knitr::kable(stat.mat)
```

	AD	CN
APP -> APOE	1625.8442483	0.2002287

	AD	CN
LRP1 -> CASP3	353.0208441	0.1203953
APP -> APBB1	26.5603763	1144.6977428
CAPN1 -> CDK5R1	406.6270082	96.3100475
LRP1 -> GSK3B	0.0671145	0.6024297
CAPN1 -> CASP3	1.6597369	181.1634912
ATP5F1 -> CASP3	0.7055343	147.6889015

Bonferroni-Holm Correction for linkage-test

```
## compute the p-values of tests after Bonferroni-Holm corrections
stat.mat.correct <- stat.mat
ps  <- as.vector(stat.mat.correct)
ps <- unlist(lapply(ps,function(o) {return(1-pchisq(o,df=1))}))
p.correct <- p.adjust(ps,"holm")
p.mat.correct <- matrix(p.correct,ncol=2,byrow = FALSE)
colnames(p.mat.correct) <- colnames(stat.mat.correct)
rownames(p.mat.correct) <- rownames(stat.mat.correct)
```

```
knitr::kable(p.mat.correct)
```

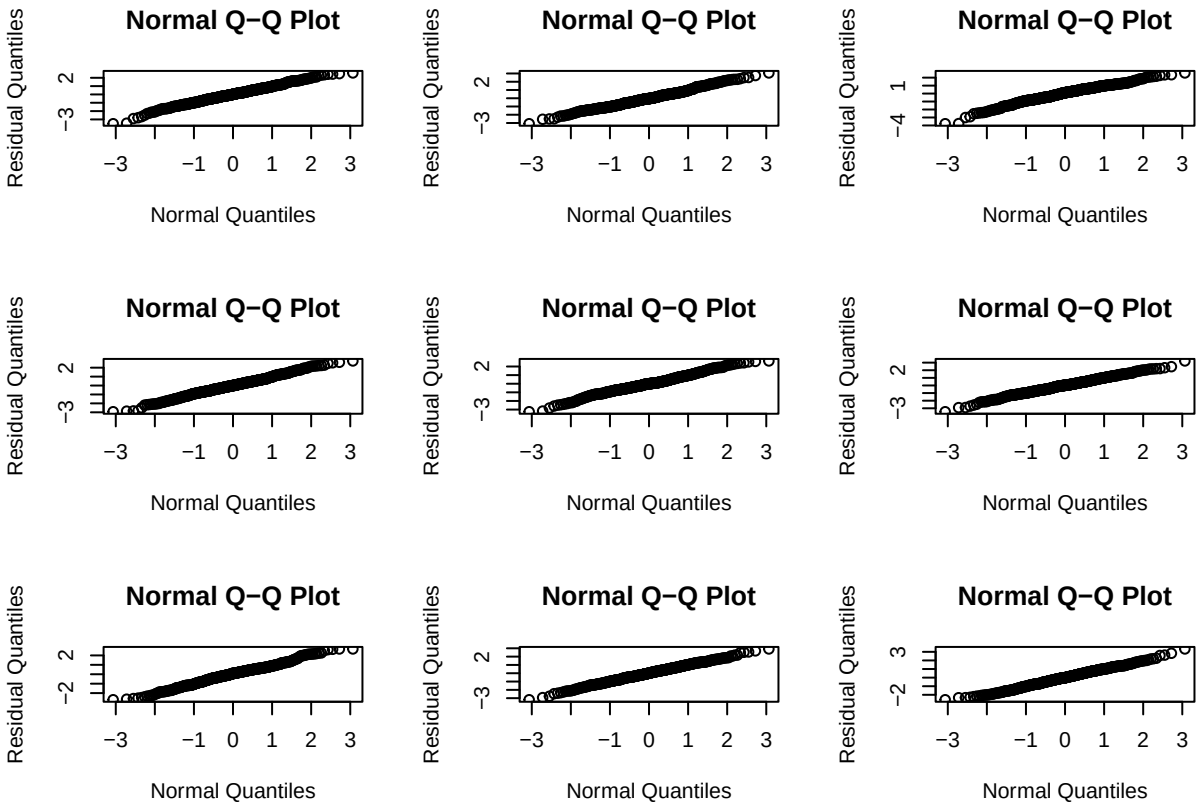
	AD	CN
APP -> APOE	0.0e+00	1
LRP1 -> CASP3	0.0e+00	1
APP -> APBB1	1.8e-06	0
CAPN1 -> CDK5R1	0.0e+00	0
LRP1 -> GSK3B	1.0e+00	1
CAPN1 -> CASP3	1.0e+00	0
ATP5F1 -> CASP3	1.0e+00	0

Normality Check

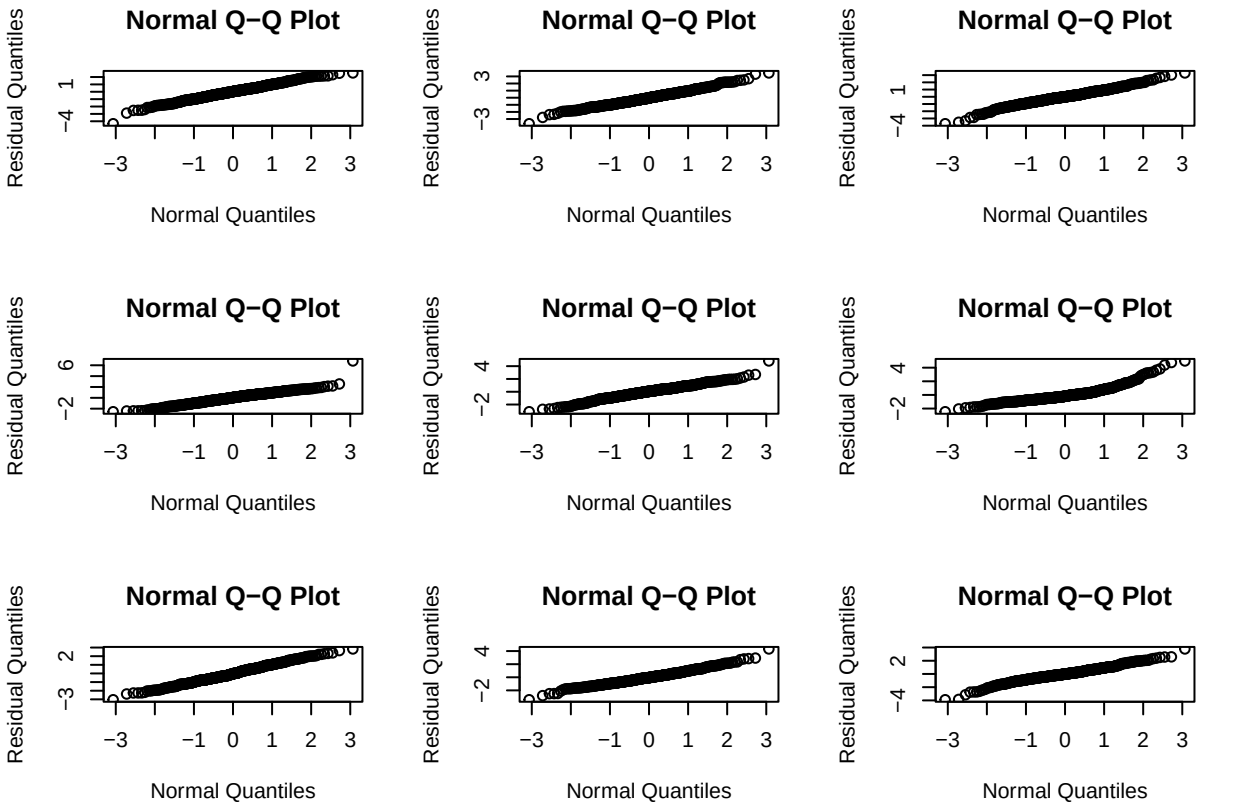
Check the normality of residuals.

AD

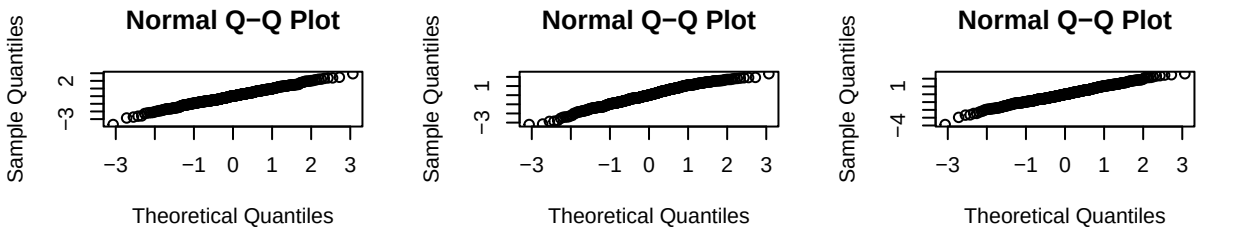
```
rmat1 <- apply(Z1,2,function(o){return(o/sd(o))})
par(mfrow=c(3,3))
for (i in 1:9){
  qqnorm(rmat1[,i],xlab = "Normal Quantiles", ylab = "Residual Quantiles")
}
```




```
par(mfrow=c(3,3))
for (i in 10:18){
  qqnorm(rmat1[,i],xlab = "Normal Quantiles", ylab = "Residual Quantiles")
}
```

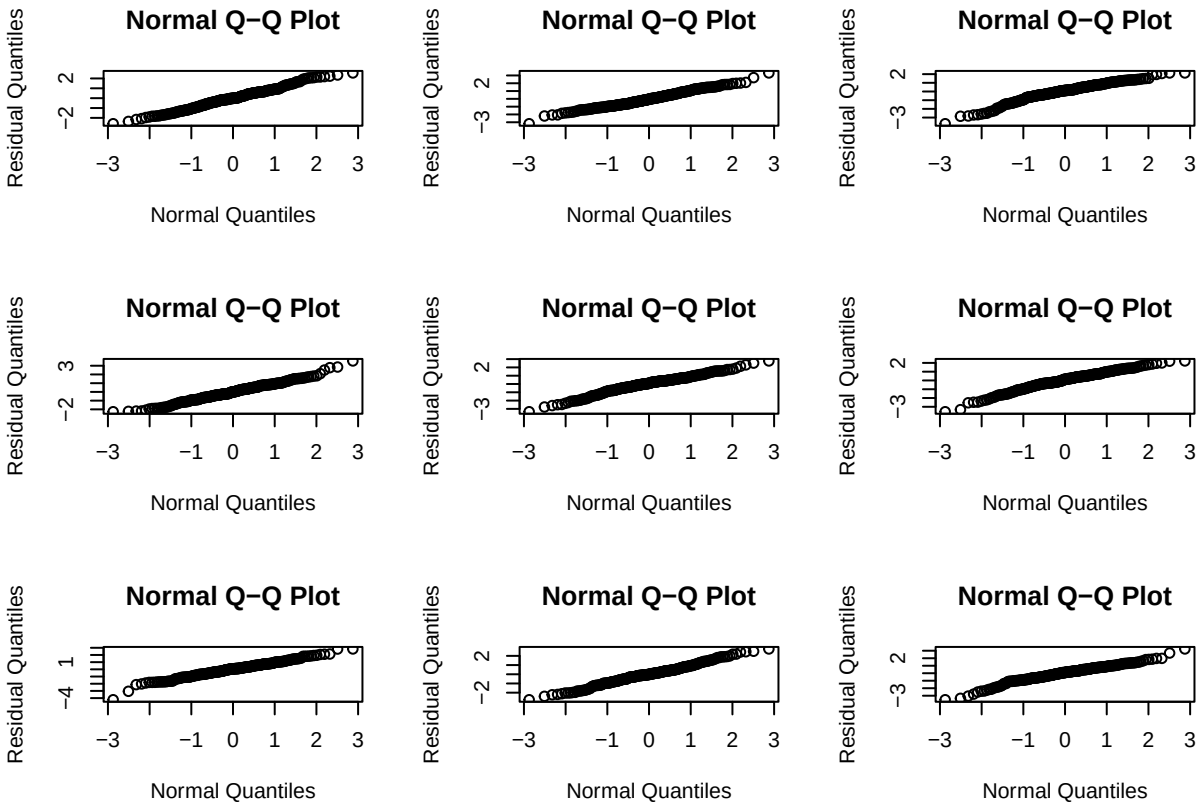



```
par(mfrow=c(3,3))
for (i in 19:21){
  qqnorm(rmat1[,i])
}
```

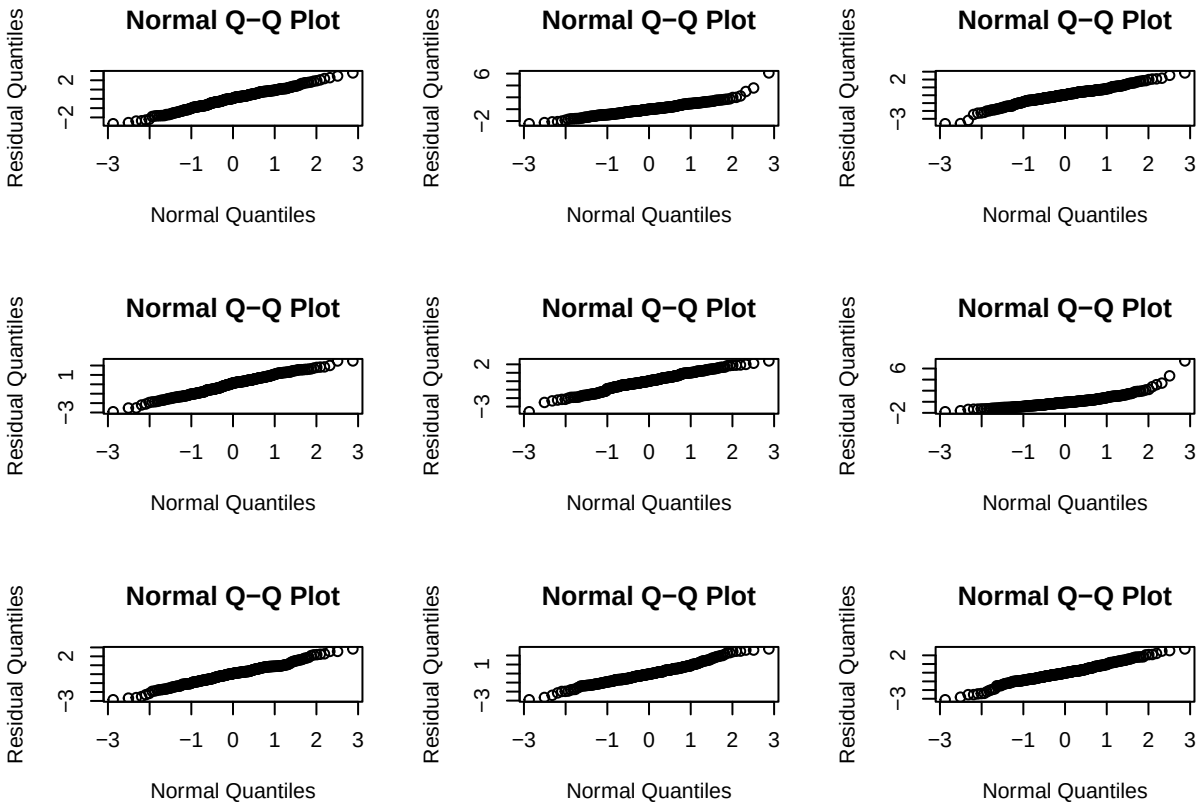


CN

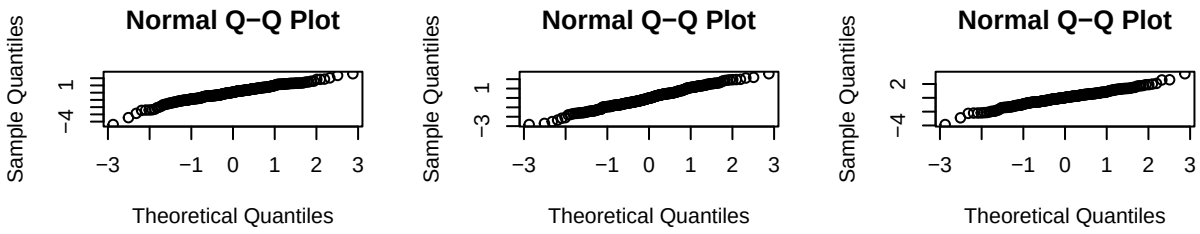
```
rmat2 <- apply(Z2,2,function(o){return(o/sd(o))})
par(mfrow=c(3,3))
for (i in 1:9){
  qqnorm(rmat2[,i],xlab = "Normal Quantiles", ylab = "Residual Quantiles")
}
```




```
par(mfrow=c(3,3))
for (i in 10:18){
  qqnorm(rmat2[,i],xlab = "Normal Quantiles", ylab = "Residual Quantiles")
}
```




```
par(mfrow=c(3,3))
for (i in 19:21){
  qqnorm(rmat2[,i])
}
```



Covariance Matrix Plot

Compute the correlations among residuals to check the existence of confounders.

```
library(ggcorrplot)
```

```
## Loading required package: ggplot2
```

```
## plot the correlation matrix of residuals for AD
corrad <- cor(rmat1)
pmatad <- cor_pmat(rmat1)
pdf(
  file ="./resi_ad.pdf",
  width = 6,
  height = 5
)
ggcorrplot(corrad,type = "lower",ggtheme = ggplot2::theme_gray,p.mat = pmatad,insig = "blank")
```

```
dev.off()
```

```
## pdf
##   2
```

```
## plot the correlation matrix of residuals for CN
corrcn <- cor(rmat2)
pmatcn <- cor_pmat(rmat2)
pdf(
  file ="./resi_cn.pdf",
  width = 6,
  height = 5
)
ggcorrplot(corrcn,type = "lower",ggtheme = ggplot2::theme_gray,p.mat = pmatcn,insig = "blank")
dev.off()
```

```
## pdf
##   2
```