```
###############################################################################
#### These are the code used to generate the figures for ANJS paper titled:
#### "Visualising the pattern of long-term genotype performance by leveraging
#### a genomic prediction model"
#### These code will generate each components of the figures.
#### These components were then assembled in power point to generate
#### a composite figure.
#### Author: Vivi Arief (v.arief1@uq.edu.au)
#### Created: 12/01/2022
##############################################################################


###############################################################################
### Some R functions
###############################################################################

########################### Gower's transformation ############################
## This part is to convert similarity matrix into disimilarity matrix
## based on Gower's equation (Gower 1966)
## simm = similarity matrix
###############################################################################
Gower.simtodis<-function(simm)
{
        diss<-matrix(0,nrow(simm),ncol(simm))
        for(i in 1:nrow(simm))
         for(j in 1:ncol(simm)){
          if(i>=j)
          {
            diss[i,j]<-simm[i,i]+simm[j,j]-2*simm[i,j]
            diss[j,i]<-diss[i,j]
          }
  }
 diss
}


#################### Average Squared Euclidean Distance #######################
## This code is for calculating average SED.
## For binary data, this is equivalent to a 1 - simple matching coefficient
## data = a two-way data table
###############################################################################
diss.mat=function(data)
{
 clust=dist(data,method="euclidean",diag=T,upper=T)
 sed.diss=as.matrix(clust^2/ncol(data))
 sed.diss
}


########################### Optimised dendrogram ##############################
## To generate an optimised dendrogram based on the Gruvaeus and Wainer
## method (1962).
## diss = a dissimilarity matrix (as.dist object)
## method = hierachical clustering method (see hclust)
```

```
###############################################################################
optimised.dendo=function(diss,method)
{
 library(gclus)
 hc=hclust(diss,method)
 hc1=reorder.hclust(hc,diss)
 hc1
}


###############################################################################
### Create biplot
### A biplot is used to display the results of the ordination analysis.
### This code might need to be adjusted to obtain a correct looking biplot.
###############################################################################

############## Ordination based on the Singular Value Decomposition ###########
### Input: A file contains the BLUP or prediction values for each genotype in
###        each environments. This file contains 3 columns:
###             GEN (genotype name or ID),
###             ENV (environment ID),
###             and BLUP
###############################################################################
## Read the data
x=read.table("BLUP.csv",header=T,row.names=NULL,stringsAsFactors=F,sep=",")
x$GEN=as.character(x$GEN)

## Create a two-way table of genotype x environment
ge.table=xtabs(x$BLUP ~ x$GEN + x$ENV)

## Replace 0 with NA. The BLUP for a genotype in non-tested environment is
zero.
## BLUP of zero indicates missing value.
ge.table[ge.table==0]=NA

## Column standardised the two-way table
x.std=scale(ge.table,center=T,scale=T)

## SVD doesn't allow missing values.
## Missing values are imputed as the mean after standardisation.
y.std=x.std
 for(i in 1:ncol(y.std))
  y.std[is.na(y.std[,i]),i]=0

## Perform the SVD
data.svd<-svd(y.std)

## Extract Singular values
d.svd<-diag(data.svd$d)

## Calculate plotting point with symmetric scaling (UD1/2; VD1/2)
# For genotypes (points)
Ysim.svd<-data.svd$u %*% sqrt(d.svd)
dimnames(Ysim.svd) <- list(dimnames(x.std)[[1]],
        paste("PC",c(1:ncol(Ysim.svd)),sep=""))
Ysim.svd <- as.data.frame(Ysim.svd)

# For environments (vectors)
Esim.svd<-data.svd$v %*% sqrt(d.svd)
dimnames(Esim.svd) <- list(dimnames(x.std)[[2]],
        paste("PC",c(1:ncol(Esim.svd)),sep=""))
Esim.svd <- as.data.frame(Esim.svd)

## Calculate % variance explained by each component
p.exp=round(diag(d.svd)^2/sum(diag(d.svd)^2)*100,2)


#### OR #####


############## Ordination based on the factor analysis #########################
### Input: File contains the scores, the loadings, and the Psi from FA model.
###        This files are obtained from the ASREML outputs
################################################################################
## Read outputs from the FA model
# Scores
Ysim.svd<-read.table("FA2_Scores.csv",sep=",",header=T,row.names=1,
        stringsAsFactors=F)
# Loadings
Esim.svd<-read.table("FA2_Loadings.csv",sep=",",header=T,row.names=1,
        stringsAsFactors=F)
# Psi
Psi <-read.table("FA2_Psi.csv",sep=",",header=T,row.names=1,
        stringsAsFactors=F)

## Calculate variance explained
ss<-svd(Esim.svd)
Lam<-as.matrix(Esim.svd) %*% ss$v
colnames(Lam)<-c(paste("XFA",1:2,sep=""))
Gvar<-Lam %*% t(Lam) + diag(Psi[,1])
varp.total<-round(mean(diag(Lam %*% t(Lam))/diag(Gvar))*100,2)
varp.fa1 <- round(mean(diag(Lam[,1] %*% t(Lam[,1]))/diag(Gvar))*100,2)
varp.fa2 <- round(mean(diag(Lam[,2] %*% t(Lam[,2]))/diag(Gvar))*100,2)
p.exp <- c(varp.fa1, varp.fa2)


###################### Plot a biplot ###########################################
### Input: plotting point for genotypes (Ysim.svd),
###        plotting points for environments (Esim.svd), and
###        the variance explained by each componenet (p.exp)
################################################################################
## Re-scaled the environmental vectors for a better fit graphs.
for(i in 1:ncol(Esim.svd))
 Esim.svd[,i] <- Esim.svd[,i]/max(abs(Esim.svd[,i]))*max(abs(Ysim.svd[,i]))

## Add additional genotypes' information for plotting.
## This information can be on the groups or other things.
## Example: GenoList.csv
## GenoList.csv contains the information of the selected genotypes.
## This file contains two columns: GEN (Genotype ID), GROUP (the group ID),
## and NAME (name to be displayed)

## Read the file for selected genotype
cv.list <-
read.table("GenoList.csv",header=T,row.names=NULL,stringsAsFactors=F,sep=",")

## Add the information of the genotype grouping the plotting data
# Add the group
Ysim.svd$Group <- cv.list$GROUP[match(dimnames(Ysim.svd)[[1]],cv.list$GEN)]

# Add the name for selected genotype
Ysim.svd$Name <- cv.list$NAME[match(dimnames(Ysim.svd)[[1]],cv.list$GEN)]

## Determined the components to be plotted
pc1 <- 1
pc2 <- 2

## Set the limit for the horizontal and the vertical axes
xmin <- min(c(Ysim.svd[,pc1],Esim.svd[,pc1]))
xmax <- max(c(Ysim.svd[,pc1],Esim.svd[,pc1]))
ymin <- min(c(Ysim.svd[,pc2],Esim.svd[,pc2]))
ymax <- max(c(Ysim.svd[,pc2],Esim.svd[,pc2]))


## Plot the genotypes as points
plot(Ysim.svd[,c(pc1,pc2)],type="p",pch=19,
     col="grey", xlim=c(xmin,xmax),ylim=c(ymin,ymax),
     xlab=paste(sub("_.*","",dimnames(Ysim.svd)[[2]][pc1]),
     " (",p.exp[pc1],"%)",sep=""),
     ylab=paste(sub("_.*","",dimnames(Ysim.svd)[[2]][pc2]),
     " (",p.exp[pc2],"%)",sep=""),
     axes=F)

## Plot the environment as vectors
for(j in 1:nrow(Esim.svd))
{
 lines(c(0,Esim.svd[j,pc1]),c(0,Esim.svd[j,pc2]),col="black",lwd=1.5)
 text(Esim.svd[j,pc1],Esim.svd[j,pc2],dimnames(Esim.svd)[[1]][j], col="black",
      lwd=1.5,cex=1.3)
}

## Add labels and group informaton for the selected genotypes
for(k in 1:nrow(Ysim.svd))

text(Ysim.svd[k,pc1],Ysim.svd[k,pc2],Ysim.svd$Name[k],col=(Ysim.svd$Group[k]+1)
,
      cex=1.3)

## Add axes
par(cex=1.2)
axis(1)
axis(2)
```

```
box()


################################################################################
### Create a dendrogram.
### The dendrogram is used to display the results of the cluter analysis.
### Cluster analysis was done for environments.
################################################################################
library(dendextend)

############### Calculate the dissimilarity matrix among environments ##########
### Input: either the standardised two-way table (x.std) or
###        the covariance matrix from FA model (Gvar)
################################################################################
## From column standardised two-way table of genotype-by-environment(x.std)
z.diss <- diss.mat(t(x.std))

### OR ###

## From the covariance matrix estimated using the FA model (Gvar)
z.sim <- cov2cor(Gvar)
# Convert the similarity matrix into a dissimilarity matrix using
# Gower's formula.
z.diss <- Gower.simtodis(z.sim)
dimnames(z.diss) <- list(dimnames(z.sim)[[1]],dimnames(z.sim)[[2]])


################### Create optimised dendrograms ###############################
### Input: dissimilarity matrix
################################################################################
z.hc=optimised.dendo(as.dist(z.diss),"ward.D")

## Plot the dendrogram
z.dend=as.dendrogram(z.hc)
par(lwd=1.5)
z.dend  %>% set("labels_cex",2) %>% set("branches_lwd", 4)  %>% plot(xlab="")


################## For comparing two dendrograms ###############################
### Input: two dendrograms (as.dendogram object)
################################################################################
dl <- dendlist(z.dend1, z.dend2)
tanglegram(dl, sort = TRUE, common_subtrees_color_lines = FALSE,
           highlight_distinct_edges  = FALSE, highlight_branches_lwd = FALSE)


################################################################################
### Create heatmap for a two-way table and a similarity matrix.
### The genotypes were ordered based on the optimised dendrogram.
################################################################################
library(fields)

################################################################################
####################### For a similarity matrix ################################
```

```
## Input: a file contains a similarity matrix. This is a squared matrix.
###############################################################################
sim <- read.table("Kmat.csv", sep=",", header=T,
row.names=1,stringsAsFactors=F)

### Ordering the matrix based on the dendogram order
## Convert the similarity matrix into a similarity matrix
# For a similarity matrix with a range of 0 to 1
diss <- 1 - sim

## OR ##

# For other similarity matrix
diss <- Gower.simtodis(sim)


## Obtain the dendrogram order
hc <- optimised.dendo(as.dist(diss),"ward.D")
line.order <- hc$label[hc$order]

## Ordered the similarity matrix based on the optimised dendrogram order
sim.ord <- sim[order(match(dimnames(sim)[[1]],line.order)),
               order(match(dimnames(sim)[[2]],line.order))]
sim.ord <- as.matrix(sim.ord)

## Replace 0 in the matrix with NA
sim.ord[sim.ord==0] <- NA

## Plot the lower-triangular of the similarity matrix
sim.ord[upper.tri(sim.ord,diag=F)]<-NA

par(mfrow=c(1,1),pty="s")
image.plot(c(1:nrow(sim.ord)),c(1:ncol(sim.ord)),sim.ord,horizontal=T, ylab="",

           xlab="",axes=F,zlim=c(0,1))
box()


################### For the two-way BLUP data #################################
### Input: a two-way genotype-by-environment table (ge.table) and
###        an dendrogram order of the genotypes (line order)
###############################################################################
## Order the genotypes based on the dendrogram order
blup.table <- ge.table[order(match(dimnames(ge.table)[[1]],line.order)),]

par(mfrow=c(1,3))
image.plot(c(1:ncol(blup.table)),c(1:nrow(blup.table)),t(blup.table),xlab="ENV"
,
           ylab="GEN", axes=F,horizontal=T)

par(cex=1.2)
axis(1)
box()
```