```
#  Supplementary material

#  Evolutionary Bioinformatics

#  How to Reveal Magnitude of Gene Signals: Hierarchical Hypergeometric
Complementary Cumulative Distribution Function

#  Bongsong Kim

########## Figure 3 ##########

rm(list=ls())

cnt   <- 1
out_1 <- 0
out_2 <- 0
out_3 <- 0

phe <- c(1:40,rep(40,420),40:79)  # Phenotype

for (cnt in 1:100){

    x <- 450
    d     <- NULL
    d     <- sample(c(0,1,2),x,replace=T)
    for (i in 1:499){
        d <- rbind(d,sample(c(0,1,2),x,replace=T))
    }
    st  <- 1
    en  <- 150
    hei <- 20
    for (i in 1:ceiling((en-st+1)/2)    ){
       d[1:floor(i*hei/((en-st+1)/2)),st+i-1] <- 1
    }
    for (i in 1:ceiling((en-st+1)/2)   ){
       d[1:floor(i*hei/((en-st+1)/2)),en-i+1] <- 1
    }
    for (i in 1:ceiling((en-st+1)/2)    ){
       d[dim(d)[1]:(dim(d)[1]-(floor(i*hei/((en-st+1)/2))+1)),st+i-1] <- 2
    }
    for (i in 1:ceiling((en-st+1)/2)   ){
       d[dim(d)[1]:(dim(d)[1]-(floor(i*hei/((en-st+1)/2))+1)),en-i+1] <- 2
    }
    st  <- 151
    en  <- 300
    hei <- 30
    for (i in 1:ceiling((en-st+1)/2)    ){
       d[1:floor(i*hei/((en-st+1)/2)),st+i-1] <- 1
    }
    for (i in 1:ceiling((en-st+1)/2)   ){
       d[1:floor(i*hei/((en-st+1)/2)),en-i+1] <- 1
    }
    for (i in 1:ceiling((en-st+1)/2)    ){
       d[dim(d)[1]:(dim(d)[1]-(floor(i*hei/((en-st+1)/2))+1)),st+i-1] <- 2
    }
    for (i in 1:ceiling((en-st+1)/2)   ){
       d[dim(d)[1]:(dim(d)[1]-(floor(i*hei/((en-st+1)/2))+1)),en-i+1] <- 2
    }
    st  <- 301
    en  <- 450
    hei <- 40
    for (i in 1:ceiling((en-st+1)/2)    ){
       d[1:floor(i*hei/((en-st+1)/2)),st+i-1] <- 1
    }
    for (i in 1:ceiling((en-st+1)/2)   ){
       d[1:floor(i*hei/((en-st+1)/2)),en-i+1] <- 1
    }
    for (i in 1:ceiling((en-st+1)/2)    ){
       d[dim(d)[1]:(dim(d)[1]-(floor(i*hei/((en-st+1)/2))+1)),st+i-1] <- 2
    }
    for (i in 1:ceiling((en-st+1)/2)   ){
       d[dim(d)[1]:(dim(d)[1]-(floor(i*hei/((en-st+1)/2))+1)),en-i+1] <- 2
    }

# HH-probability

  len <- dim(d)[2]
  p_val1 <- 0
  p_val2 <- 0
  for (z in 1:len){
    gen    <- 0
    x      <- 0
    len_x  <- 0
    bot    <- 0
    mid    <- 0
    top    <- 0
    succ_1 <- 0
    succ_2 <- 0
    gen <- d[,z]
    len_x[1] <- length(which(gen==0))
    len_x[2] <- length(which(gen==1))
    len_x[3] <- length(which(gen==2))
    x[1] <-  mean(phe[which(gen==0)])
    x[2] <-  mean(phe[which(gen==1)])
    x[3] <-  mean(phe[which(gen==2)])
    if (x[1] == x[2]){ next }
    if (x[1] == x[3]){ next }
    if (x[2] == x[3]){ next }
    bot  <- which(rank(x) == 1)
    mid  <- which(rank(x) == 2)
    top  <- which(rank(x) == 3)
    order(phe)[(len_x[bot]+len_x[mid]+1):(len_x[bot]+len_x[mid]+len_x[top])] -> k1
    succ_1 <- which( gen[k1] == c(0,1,2)[top])
    order(phe)[1:len_x[bot]] -> k2
    succ_2 <- which( gen[k2] == c(0,1,2)[bot])
    su_1 <- 0
    for (i in 0:length(succ_1)){
      su_1 <- su_1 + dhyper(i,len_x[top],len_x[bot]+len_x[mid],len_x[top])
    }
    su_2 <- 0
    for (i in 0:length(succ_2)){
      su_2 <- su_2 + dhyper(i,len_x[bot],len_x[mid]+len_x[top],len_x[bot])
    }
    p_val1[z] <- 1-sqrt((su_1)*(su_2))
  }
  out_1 <- out_1 + p_val1

# F test
  vec <- 0
  res1 <- 0
  d <- data.frame(d)
  for (i in 1:len){
    res1 <- summary(lm(phe ~ 1 + d[,i],data = d))
    vec[i] <- res1$coefficients[2,4]
  }
  out_2 <- out_2 + vec

# HA-coefficient
  res <- 0
  for (i in 1:len){
   su  <- 0
   geno  <- d[,i]
   ave_0 <- mean( phe[geno == 0])
   ave_1 <- mean( phe[geno == 1])
   ave_2 <- mean( phe[geno == 2])
   ord <- c(ave_0, ave_1, ave_2)
   ord <- rank(ord)
   ref <- c(0,1,2)
   who_len <- length(geno)
   min_len <- length( which( geno == ref[ which(ord == 1) ]) )
   mid_len <- length( which( geno == ref[ which(ord == 2) ]) )
   max_len <- length( which( geno == ref[ which(ord == 3) ]) )
   i_list <- sort(phe,decreasing=F)
   top_g0 <- i_list[ 1 : min_len ]
   top_g1 <- i_list[ (min_len + 1) : (min_len+mid_len) ]
   top_g2 <- i_list[ (min_len+mid_len + 1) : (min_len + mid_len + max_len) ]
   r_list <- sort( phe,decreasing=T)
   bot_g0 <- r_list[ 1 : min_len ]
   bot_g1 <- r_list[ (min_len + 1) : (min_len+mid_len) ]
   bot_g2 <- r_list[ (min_len+mid_len + 1) : (min_len + mid_len + max_len) ]
   top_s0 <- sum(top_g0)
   top_s1 <- sum(top_g1)
   top_s2 <- sum(top_g2)
   bot_s0 <- sum(bot_g0)
   bot_s1 <- sum(bot_g1)
   bot_s2 <- sum(bot_g2)
   obs_s0 <- sum( phe[which( geno == ref[ which(ord == 1) ])] )
   obs_s1 <- sum( phe[which( geno == ref[ which(ord == 2) ])] )
   obs_s2 <- sum( phe[which( geno == ref[ which(ord == 3) ])] )
   su <- sum(obs_s0,obs_s1,obs_s2)
   x2 <- (su*log(obs_s1 + obs_s2) - (obs_s1 + obs_s2)) - (su*log(bot_s1+bot_s2) -
(bot_s1+bot_s2))
   x1 <- (su*log(top_s1 + top_s2) - (top_s1 + top_s2)) - (su*log(bot_s1+bot_s2) -
(bot_s1+bot_s2))
   obs_1 <- x2/x1
   x2 <- (su*log(obs_s2) - obs_s2) - (su*log(bot_s2) - bot_s2)
   x1 <- (su*log(top_s2) - top_s2) - (su*log(bot_s2) - bot_s2)
   obs_2 <- x2/x1
   res[i] <-(obs_1*obs_2)^0.5
  }
  out_3 <- out_3 + res
}

###### Figure 4

  par(mfrow=c(1,3))
  plot(-log10(out_1/cnt),pch=19,xlab="",ylab="-log10(P value)")
  title(main = "HH-CCDF")
  plot(-log10(out_2/cnt),pch=19,xlab="",ylab="-log10(P value)")
  title(main = "F test")
  plot(out_3/cnt,pch=19,xlab="",ylab="HA-coefficient")
  title(main = "HA-coefficient algorithm")



########## Figures 5 and 6 ##########

install.packages("RCurl")
require(RCurl)
d   <-
read.csv(text=getURL("https://raw.githubusercontent.com/bongsongkim/HH_CCDF/master/r
ice_genotype.csv"),row.names=1)
phe <-
read.csv(text=getURL("https://raw.githubusercontent.com/bongsongkim/HH_CCDF/master/r
ice_phenotype.csv"),header=T)
phe <- phe[,2]

# HH-probability

  len    <- dim(d)[2]
  p_val1 <- 0
  p_val2 <- 0
  for (z in 1:len){
    gen    <- 0
    x      <- 0
    len_x  <- 0
    bot    <- 0
    mid    <- 0
    top    <- 0
    succ_1 <- 0
    succ_2 <- 0
    gen <- d[,z]
    len_x[1] <- length(which(gen==0))
    len_x[2] <- length(which(gen==1))
    len_x[3] <- length(which(gen==2))
    x[1] <-  mean(phe[which(gen==0)])
    x[2] <-  mean(phe[which(gen==1)])
    x[3] <-  mean(phe[which(gen==2)])
    if (x[1] == x[2]){ next }
    if (x[1] == x[3]){ next }
    if (x[2] == x[3]){ next }
    bot  <- which(rank(x) == 1)
    mid  <- which(rank(x) == 2)
    top  <- which(rank(x) == 3)
    order(phe)[(len_x[bot]+len_x[mid]+1):(len_x[bot]+len_x[mid]+len_x[top])] -> k1
    succ_1 <- which( gen[k1] == c(0,1,2)[top])
    order(phe)[1:len_x[bot]] -> k2
    succ_2 <- which( gen[k2] == c(0,1,2)[bot])
    su_1 <- 0
    for (i in 0:length(succ_1)){
      su_1 <- su_1 + dhyper(i,len_x[top],len_x[bot]+len_x[mid],len_x[top])
    }
    su_2 <- 0
    for (i in 0:length(succ_2)){
      su_2 <- su_2 + dhyper(i,len_x[bot],len_x[mid]+len_x[top],len_x[bot])
    }
    p_val1[z] <- sqrt(( 1-su_1)*(1-su_2))
  }
```

```
# F test
  vec <- 0
  res1 <- 0
  len    <- dim(d)[2]
  d <- data.frame(d)
  for (i in 1:len){
    res1 <- summary(lm(phe ~ 1 + d[,i],data = d))
    vec[i] <- res1$coefficients[2,4]
  }

# HA-coefficient

  phe = max(phe) - min(phe) + phe
  res <- 0
  for (i in 1:len){

   su  <- 0
   geno  <- d[,i]
   ave_0 <- mean( phe[geno == 0])
   ave_1 <- mean( phe[geno == 1])
   ave_2 <- mean( phe[geno == 2])
   ord <- c(ave_0, ave_1, ave_2)
   ord <- rank(ord)
   ref <- c(0,1,2)
   who_len <- length(geno)
   min_len <- length( which( geno == ref[ which(ord == 1) ]) )
   mid_len <- length( which( geno == ref[ which(ord == 2) ]) )
   max_len <- length( which( geno == ref[ which(ord == 3) ]) )
   i_list <- sort(phe,decreasing=F)
   top_g0 <- i_list[ 1 : min_len ]
   top_g1 <- i_list[ (min_len + 1) : (min_len+mid_len) ]
   top_g2 <- i_list[ (min_len+mid_len + 1) : (min_len + mid_len + max_len) ]
   r_list <- sort( phe,decreasing=T)
   bot_g0 <- r_list[ 1 : min_len ]
   bot_g1 <- r_list[ (min_len + 1) : (min_len+mid_len) ]
   bot_g2 <- r_list[ (min_len+mid_len + 1) : (min_len + mid_len + max_len) ]
   top_s0 <- sum(top_g0)
   top_s1 <- sum(top_g1)
   top_s2 <- sum(top_g2)
   bot_s0 <- sum(bot_g0)
   bot_s1 <- sum(bot_g1)
   bot_s2 <- sum(bot_g2)
   obs_s0 <- sum( phe[which( geno == ref[ which(ord == 1) ])] )
   obs_s1 <- sum( phe[which( geno == ref[ which(ord == 2) ])] )
   obs_s2 <- sum( phe[which( geno == ref[ which(ord == 3) ])] )
   su <- sum(obs_s0,obs_s1,obs_s2)
   x2 <- (su*log(obs_s1 + obs_s2) - (obs_s1 + obs_s2)) - (su*log(bot_s1+bot_s2) -
(bot_s1+bot_s2))
```

```
   x1 <- (su*log(top_s1 + top_s2) - (top_s1 + top_s2)) - (su*log(bot_s1+bot_s2) - 
(bot_s1+bot_s2))
   obs_1 <- x2/x1
   x2 <- (su*log(obs_s2) - obs_s2) - (su*log(bot_s2) - bot_s2)
   x1 <- (su*log(top_s2) - top_s2) - (su*log(bot_s2) - bot_s2)
   obs_2 <- x2/x1
   res[i] <-(obs_1*obs_2)^0.5
  }

###### Figure 5

  par(mfrow=c(3,1))
  plot(-log10(p_val1),type="l",col="red",ylab="-log10(P value)",ann=FALSE,cex.lab=2, 
cex.axis=2)
  plot(-log10(vec),type="l",col="blue",ylab="-log10(P value)",ann=FALSE,cex.lab=2, 
cex.axis=2)
  plot(res,type="l",col="green",ylab="HA-coefficient",ann=FALSE,cex.lab=2, 
cex.axis=2)

###### Figure 6

  std1 <- (-log10(p_val1) - min(-log10(p_val1)))/(max(-log10(p_val1)) - 
min(-log10(p_val1)))
  std2 <- (-log10(vec) - min(-log10(vec)))/(max(-log10(vec)) - min(-log10(vec)))
  std3 <- (res - min(res))/(max(res) - min(res))

  par(mfrow=c(3,1))
  plot(std1,type="l",col="red",lty=1,ann=FALSE,cex.lab=2, cex.axis=2)
  par(new=TRUE)
  plot(std2,type="l",col="blue",lty=1,ann=FALSE,cex.lab=2, cex.axis=2)

  plot(std1,type="l",col="red",lty=1,ann=FALSE,cex.lab=2, cex.axis=2)
  par(new=TRUE)
  plot(std3,type="l",col="green",lty=1,ann=FALSE,cex.lab=2, cex.axis=2)

  plot(std2,type="l",col="blue",lty=1,ann=FALSE,cex.lab=2, cex.axis=2)
  par(new=TRUE)
  plot(std3,type="l",col="green",lty=1,ann=FALSE,cex.lab=2, cex.axis=2)
```