**Script for Haplotype analysis**

**Script for Haplotype analysis**

library(ape)

library(pegas)

#load data (FASTA files)

seq<- read.dna("All\_333\_AMA1\_sequences\_yearwise.fasta", format="fasta")

#Get and Write haplotype sequences

GetHaplo("../../Popgendata\_BegCape\_combined.fa", align = NA, saveFile = TRUE, outname = "194\_Haplotypes\_seq\_res.fa", format = "fasta",seqsNames = NA, silent = TRUE)

GetHaplo("../../Popgendata\_BegCape\_combined.fa", align = NA, saveFile = TRUE, outname = "", format = "fasta",seqsNames = NA, silent = TRUE)

#check loaded data

seq

#convert the DNA sequence data into haplotypes

AMA1\_Haplotypes <- haplotype(seq)

AMA1\_Haplotypes

#print all haplotypes

summary(AMA1\_Haplotypes)

AMA1\_Haplotypes

#Change rownames of haplotypes

row.names(AMA1\_Haplotypes) <- 1 : 194 #look at no accordingly change it

rownames(AMA1\_Haplotypes) <- paste0('hap', rownames(AMA1\_Haplotypes))

#print all haplotypes

summary(AMA1\_Haplotypes)

#Extract mutations with position for all the haplotypes

all\_haplotypes\_diff <- as.data.frame(diffHaplo(AMA1\_Haplotypes,1:194)) # no. can be changed after checking how many total haplotypes are there

#Write in a file

write.table(all\_haplotypes\_diff, file= "all\_haplotypes\_yearwise\_changes.txt")

#Create haplotype network

AMA1\_Net <- haploNet(AMA1\_Haplotypes)

AMA1\_Net2 <- mjn(AMA1\_Haplotypes)

#Plot haplotype network

#Note: Use fast = TRUE, just to explore the data

plot(AMA1\_Net, size = attr(AMA1\_Net, "freq"), fast = TRUE, labels=F, show.mutation=2)

# Note: use fast = FALSE, because the structure of the resulting plot is less messy

plot(AMA1\_Net, size = attr(AMA1\_Net, "freq"), fast = FALSE)

#Next to know which samples belong to which haplotype

ind.hap<- with(stack(setNames(attr(AMA1\_Haplotypes, "index"), rownames(AMA1\_Haplotypes))), table(hap=ind, pop=rownames(seq)[values]))

#Print haplotype list

ind.hap

data <-t(ind.hap)

#Barplot

head(ind.hap)

ind.hap

data <- as.matrix(ind.hap)

data

names<- factor(rownames(data))

hap\_names <- unique(names)

par(mar = c(4, 3, 3, 6), xpd = TRUE)

barplot(ind.hap, las=2, cex.axis=0.5, cex.names = 0.5, col=rainbow(length(hap\_names)))

legend("right", inset = c(- 0.1, 2), legend = hap\_names, pch = 7, col = rainbow(length(hap\_names)), cex = 0.5)

library(reshape)

library(ggplot2)

melted\_df <- melt(ind.hap)

head(melted\_df)

colnames(melted\_df) <- c("Haplotype", "Population", "Number")

#Plot with counts

p <- ggplot(melted\_df, aes(x = Population, y = Number)) +geom\_col(aes(fill = Haplotype, width = 0.9))

p+ theme(legend.key.size = unit(0.6, "cm"), legend.text = element\_text( color="Black", size=9), axis.text.x = element\_text( color="Black", size=10, angle=90), axis.text.y = element\_text( color="Black", size=10, angle=90))

Value= melted\_df$Number

Percentage= round(Value / sum(Value) \* 100,2)

#Plot with percentage

pp <- ggplot(melted\_df, aes(x = Population, y = Percentage, fill = Haplotype)) + geom\_bar(position = "fill",stat = "identity") + scale\_y\_continuous(labels = scales::percent\_format())

pp + theme(legend.key.size = unit(0.6, "cm"), legend.text = element\_text( color="Black", size=9), axis.text.x = element\_text( color="Black", size=10, angle=90), axis.text.y = element\_text( color="Black", size=10))

#Write into a file

write.table(ind.hap, file="individual\_sample\_haplotype\_yearwise.txt")

#Next to convert based on region/loacation

#convert into dataframe

mydata <- as.data.frame(ind.hap)

hap\_list<- mydata[mydata$Freq == 1,]

hap\_list

# let create haplotypes based on locations, here, string split the names by underscores.

locations <- strsplit(as.character(hap\_list$pop), "")

locations

# Next extract first item in each list

locations1 <- sapply(locations, "[[", 1)

head(locations1)

#Now make a table with our new locations list and the corresponding haplotypes

new.hap <- table(hap\_list$hap, locations1)

new.hap

#Draw a barplot

barplot(new.hap, main= "No of.Haplotypes of AMA1 in two regions", cex.axis = 0.8, cex.lab=0.9, cex.names=0.8, ylab="Number of Haplotypes", xlab="Location", space=0.2)

#Write into a file

write.table(new.hap, file="group\_haplotype.txt")

#Create haplotype network

#Create haplotype network (quickly), but it woukd be messy without labels

plot(AMA1\_Net, size = attr(AMA1\_Net, "freq"), fast = T, show.mutation=2, labels=F, cex = 0.5, main = "Haplotype network for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

#Create haplotype network (quickly), but it woukd be messy with labels of haplotypes

plot(AMA1\_Net, size = attr(AMA1\_Net, "freq"), fast = T, show.mutation=2, labels=T, cex = 0.5, main = "Haplotype network for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

######### Select frequency criteria to plot

haps1 <- subset(AMA1\_Haplotypes, minfreq = 1)

haps2 <- subset(AMA1\_Haplotypes, minfreq = 2)

haps3 <- subset(AMA1\_Haplotypes, minfreq = 3)

############################# Freuqncy 1 ###########

haps1 <- subset(AMA1\_Haplotypes, minfreq = 1)

#Next to know which samples belong to which haplotype

# Note: Change the "haps" value according to the frequncy of haplotypes you want choose for network

haps\_sort1 <- sort(haps1)

ind.hap<- with(stack(setNames(attr(haps1, "index"), rownames(haps1))), table(hap=ind, pop=rownames(seq)[values]))

#convert it into matrix

indiv\_haplotypes <- as.matrix(ind.hap)

#barplot(indiv\_haplotypes, las=2)

#Write in file

#write.table(indiv\_haplotypes, file ="indiv\_hapltype.txt")

#create group or location-wise table

#convert into dataframe

mydata <- as.data.frame(ind.hap)

hap\_list<- mydata[mydata$Freq == 1,]

hap\_list

# let create haplotypes based on locations, here, string split the names by underscores.

locations <- strsplit(as.character(hap\_list$pop), "\_")

locations

# Next extract first item in each list

locations1 <- sapply(locations, "[[", 1)

head(locations1)

#Now make a table with our new locations list and the corresponding haplotypes

new.hap <- table(hap\_list$hap, locations1)

new.hap

#write.table(new.hap, file="new\_haplotypes.txt")

#Create haplotype network

haps\_sort <- sort(haps1)

network <- haploNet(haps1, getProb = TRUE)

network11 <- haploNet(haps\_sort)

#Plot without labels of samples, to quickly (fast=T), probably messy network

plot(network, size = attr(network, "freq"), fast = T, show.mutation=2, labels=F, cex = 0.5, main = "Haplotype network (freq >= 1) for Begro and Capecoast", pie=new.hap, lwd = 0.1, scale=1)

#Plot without labels of samples

plot(network1, size = attr(network1, "freq"), fast = T, show.mutation=2, labels=T, cex = 0.5, main = "Haplotype network (freq >= 2) for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

#Plot proper network (fast=F) without labels

plot(network1, size = attr(network1, "freq"), fast = F, show.mutation=2, labels=F, cex = 0.5, main = "Haplotype network (freq >= 2) for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

#Plot proper network (fast=F) with labels

plot(network1, size = attr(network1, "freq"), fast = F, show.mutation=2, labels=T, cex = 0.5, main = "Haplotype network (freq >= 2) for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

#Plot proper network (fast=F) with labels

plot(network22, size = attr(network22, "freq"), fast = F, show.mutation=2, labels=T, cex = 0.5, main = "Haplotype network (freq >= 2) for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

#Add legends

legend(180,40, colnames(new.hap), col=rainbow(ncol(new.hap)), pch=20)

########################### Freuqncy 2 ################

haps2 <- subset(AMA1\_Haplotypes, minfreq = 2)

#Next to know which samples belong to which haplotype

# Note: Change the "haps" value according to the frequncy of haplotypes you want choose for network

#haps\_sort2 <- sort(haps2)

#haps = haps\_sort2

ind.hap<- with(stack(setNames(attr(haps2, "index"), rownames(haps2))), table(hap=ind, pop=rownames(seq)[values]))

#convert it into matrix

indiv\_haplotypes <- as.matrix(ind.hap)

#barplot(indiv\_haplotypes, las=2)

#Write in file

#write.table(indiv\_haplotypes, file ="indiv\_hapltype.txt")

#create group or location-wise table

#convert into dataframe

mydata <- as.data.frame(ind.hap)

hap\_list<- mydata[mydata$Freq == 1,]

hap\_list

# let create haplotypes based on locations, here, string split the names by underscores.

locations <- strsplit(as.character(hap\_list$pop), "\_")

locations

# Next extract first item in each list

locations1 <- sapply(locations, "[[", 1)

head(locations1)

#Now make a table with our new locations list and the corresponding haplotypes

new.hap <- table(hap\_list$hap, locations1)

new.hap

#write.table(new.hap, file="new\_haplotypes.txt")

#Create haplotype network

haps\_sort2 <- sort(haps2)

network <- haploNet(haps, getProb = TRUE)

network1 <- haploNet(haps2)

network22 <- haploNet(haps\_sort2)

#Plot without labels of samples, to quickly (fast=T), probably messy network

plot(network1, size = attr(network1, "freq"), fast = T, show.mutation=2, labels=F, cex = 0.5, main = "Haplotype network (freq >= 2) for Begro and Capecoast", pie=new.hap, lwd = 0.1, scale=1)

#Plot without labels of samples

plot(network1, size = attr(network1, "freq"), fast = T, show.mutation=2, labels=T, cex = 0.5, main = "Haplotype network (freq >= 2) for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

#Plot proper network (fast=F) without labels

plot(network1, size = attr(network1, "freq"), fast = F, show.mutation=2, labels=F, cex = 0.5, main = "Haplotype network (freq >= 2) for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

png(

"test.png",

width = 10,

height = 10,

units = "in",

res = 150,

pointsize = 1

)

par(

mar = c(5, 5, 2, 2),

xaxs = "i",

yaxs = "i",

cex.axis = 10,

cex.main = 20,

cex.lab = 10

)

plot(network1, size = attr(network1, "freq"), fast = F, show.mutation=2, labels=T, cex = 10, main = "", pie=new.hap, lwd = 0.5, scale=1)

legend(-50,10, colnames(new.hap), col=rainbow(ncol(new.hap)), cex=10, pch=20)

dev.off()

#Plot proper network (fast=F) with labels

plot(network1, size = attr(network1, "freq"), fast = F, show.mutation=2, labels=T, cex = 0.5, main = "Haplotype network (freq >= 2) for Begoro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

#Add legends

#Add legends

legend(-130,10, colnames(new.hap), col=rainbow(ncol(new.hap)), pch=20)

dev.off()

legend(180,40, colnames(new.hap), col=rainbow(ncol(new.hap)), pch=20)

#Plot proper network (fast=F) with labels

plot(network22, size = attr(network22, "freq"), fast = F, show.mutation=2, labels=T, cex = 0.5, main = "Haplotype network (freq >= 2) for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

############ for Frequency 3

haps3 <- subset(AMA1\_Haplotypes, minfreq = 3)

#Next to know which samples belong to which haplotype

# Note: Change the "haps" value according to the frequncy of haplotypes you want choose for network

haps3

haps\_sort3 <- sort(haps3)

ind.hap<- with(stack(setNames(attr(haps3, "index"), rownames(haps3))), table(hap=ind, pop=rownames(seq)[values]))

#convert it into matrix

indiv\_haplotypes <- as.matrix(ind.hap)

#barplot(indiv\_haplotypes, las=2)

#Write in file

#write.table(indiv\_haplotypes, file ="indiv\_hapltype.txt")

#create group or location-wise table

#convert into dataframe

mydata <- as.data.frame(ind.hap)

hap\_list<- mydata[mydata$Freq == 1,]

hap\_list

# let create haplotypes based on locations, here, string split the names by underscores.

locations <- strsplit(as.character(hap\_list$pop), "\_")

locations

# Next extract first item in each list

locations1 <- sapply(locations, "[[", 1)

head(locations1)

#Now make a table with our new locations list and the corresponding haplotypes

new.hap <- table(hap\_list$hap, locations1)

new.hap

#write.table(new.hap, file="new\_haplotypes.txt")

#Create haplotype network

haps\_sort <- sort(haps1)

network3 <- haploNet(haps3)

network4 <- mjn(haps3)

network33 <- haploNet(haps\_sort3)

#Plot without labels of samples, to quickly (fast=T), probably messy network

plot(network3, size = attr(network3, "freq"), fast = T, show.mutation=2, labels=F, cex = 0.5, main = "Haplotype network (freq >= 3) for Begro and Capecoast", pie=new.hap, lwd = 0.1, scale=1)

plot(network4, fast = T, show.mutation=2, labels=F, cex = 0.5, main = "Haplotype network (freq >= 3) for Begro and Capecoast", pie=new.hap, lwd = 0.1, scale=1)

#Plot without labels of samples

plot(network3, size = attr(network3, "freq"), fast = T, show.mutation=2, labels=T, cex = 0.5, main = "Haplotype network (freq >= 3) for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

#Plot proper network (fast=F) without labels

plot(network3, size = attr(network3, "freq"), fast = F, show.mutation=2, labels=F, cex = 0.5, main = "Haplotype network (freq >= 3) for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

#Plot proper network (fast=F) with labels

plot(network3, size = attr(network3, "freq"), fast = F, show.mutation=2, labels=T, cex = 0.5, main = "Haplotype network (freq >= 3) for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

#Plot proper network (fast=F) with labels

plot(network33, size = attr(network33, "freq"), fast = F, show.mutation=2, labels=T, cex = 0.5, main = "Haplotype network (freq >= 3) for Begro and Capecoast", pie=new.hap, lwd = 0.5, scale=1)

#Add legends

legend(180,40, colnames(new.hap), col=rainbow(ncol(new.hap)), pch=20)

**Script for mutations analysis**

#load libraries

library(gplots)

library(dplyr) #required for modification of data

library(pryr) #required to save basic rplots as object

library(ggplot2)

library(gt)

library(ggpubr)

library(gridExtra)

msa\_sel1 <- read.table("Mutations\_AMA1-Data.txt", header=TRUE, sep="\t")

dim(msa\_sel1)

head(msa\_sel1)

pos <- msa\_sel1[1]

ref<- msa\_sel1[2]

msa\_sel <- msa\_sel1 #remove the "POS" column

#Report on the number of mutations in each sample:

x <- ncol(msa\_sel)

x

y <- nrow(msa\_sel)

y

msa\_sel

#create an empty data frame to store the new data

df\_new1 <- data.frame(matrix(0, ncol = x+1, nrow = y))

colnames(df\_new1) <- colnames(msa\_sel)

head(msa\_sel)

#define i value

i=3

#Create a for loop, to assign variant (i.e. nucleotide mutated to which new variant/nucleotide). Here, we assigning “-” if there is no mutation, else assigning to mutation

for (i in 2:x) {

df\_new1[,i] <- ifelse(msa\_sel[,2]== msa\_sel[,i],"-", paste0(msa\_sel[,i]))

}

df\_new1

#Load library

library(dplyr) #required for modification of data

#Remove reference genome column and last column (NA column) from the dataframe

df\_new2 <- select(df\_new1,-1, -ncol(df\_new1))

#Add position column

final\_df <- cbind(ref, df\_new2)

final\_df

row.names(final\_df) <- msa\_sel1 [,1]

head(final\_df)

final\_df %>% gt()

#summary table

main.title <- paste("Summary of mutations variants in Haplotypes of AMA1")

mutation\_summary <- ggtexttable(final\_df, rows = NULL, theme = ttheme("light", padding = unit(c(11, 2.5), "mm"), tbody.style = tbody\_style(fill = "white", size = 9), colnames.style = colnames\_style(fill = "white", size = 10, color = "Black", rot=90)))

mutation\_summary1 <- mutation\_summary %>% tab\_add\_title(text = main.title, face = "bold", padding = unit(5, "line"))

mutation\_summary1

Protein <- 'AMA1\_haplotypes\_Variants.txt'

report\_name <- toString(sub(".txt", ".pdf", Protein))

grid <- grid.arrange(mutation\_summary1, nrow=1, ncol=1)

ggsave(grid, file=report\_name, height = 30 , width = 30, device = pdf, dpi = 300)

#Generate MSA variant plot

#Transpose

new\_dfT <- t(final\_df)

write.table(new\_dfT, file= "AMA1\_Haplotypes\_variant\_info.txt", row.names = T, sep ="\t")

library(reshape2)

melted\_mat <- melt(new\_dfT)

head(melted\_mat)

#draw MSA the plot

plot1 <- ggplot(melted\_mat) + #add data and fill cells

geom\_tile(data = melted\_mat, aes(x=factor(Var1), y=factor(Var2), fill=value), colour = "black") + #add black border around cells

geom\_text(data = melted\_mat, aes(x=factor(Var1), y=factor(Var2),label = value), size=2, family = "Helvetica") +

scale\_fill\_manual(values = c("-" = "lightgray", "A" = "lightgreen", "C" = "pink", "G" = "lightblue", "T" = "yellow", "V"="red", "\*"="white")) +

#coord\_equal() +

ylab("Nucleotide Position") +

xlab("Haplotypes") +

labs(fill = "NT") +

labs(title= paste("Number of variants in the Haplotypes for AMA1:" , ncol(new\_dfT) )) +

theme(axis.text = element\_text(size=6, family = "Helvetica"),

legend.title = element\_text(color = "black", size = 6),

legend.text = element\_text(color = "black"),

axis.title = element\_text(size=8, vjust = 2, face="italic"),

axis.text.x = element\_text(angle = 90, vjust = 0.5, size = 6, color = "black"),

axis.text.y = element\_text(size = 6, color = "black"),

panel.border = element\_rect(colour = "black", fill=NA, size=1)

)

#make the first plot

plot(plot1)

Protein <- 'AMA1\_haplotypes\_MSA.txt'

report\_name <- toString(sub(".txt", ".pdf", Protein))

grid <- grid.arrange(plot1, nrow=1, ncol=1)

ggsave(grid, file=report\_name, height = 30 , width = 30, device = pdf, dpi = 300)

########

head(new\_dfT)

#Report on the number of mutations in each sample:

x <- ncol(msa\_sel)

x

y <- nrow(msa\_sel)

y

#create an empty data frame to store the new data

df\_mut <- data.frame(matrix(0, ncol = x+1, nrow = y))

colnames(df\_mut) <- colnames(msa\_sel)

#define i value

i=3

#Create a for loop, to assign variant (i.e. nucleotide mutated to which new variant/nucleotide). Here, we assigning “-” if there is no mutation, else assigning to mutation

for (i in 2:x) {

df\_mut[,i] <- ifelse(msa\_sel[,2]== msa\_sel[,i],0,1)

}

#Load library

library(dplyr) #required for modification of data

#Remove reference genome column and last column (NA column) from the dataframe

df\_mut <- select(df\_mut, -1, -ncol(df\_mut))

#df\_mut%>% gt()

head(df\_mut)

#Compute sum of mutations per samples

summ1 <- as.data.frame(colSums(df\_mut))

#Samples

samples<- as.data.frame(row.names(summ1))

#Combine samples and sum

summ2 <- cbind(samples, summ1)

#Add column names

colnames(summ2) <- c("Sample", "Mutations\_Number")

head(summ2)

#make a barplot

bb <- ggplot(summ2, aes(x = Sample, y = Mutations\_Number)) +geom\_col(aes(width = 0.8))

bb+ theme(legend.key.size = unit(0.6, "cm"), legend.text = element\_text( color="Black", size=9), axis.text.x = element\_text( color="Black", size=10, angle=90), axis.text.y = element\_text( color="Black", size=10, angle=90))

bb <- ggplot(summ2, aes(x=Sample, y=Mutations\_Number)) + geom\_col(stat="identity", fill="steelblue", width = 0.5) +

labs(title= paste("No. of mutations per Haplotype" )) +

theme(axis.text = element\_text(size=10),

legend.title = element\_text(color = "black", size = 8),

legend.text = element\_text(color = "black"),

axis.title = element\_text(size=8, vjust = 2, face="italic"),

axis.text.x = element\_text(color = "black", angle = 90, vjust = 0.5, size = 8),

axis.text.y = element\_text(color = "black",size = 8, angle = 90,vjust = 0.5),

panel.border = element\_rect(colour = "black", fill=NA, size=1))

bb

Protein <- 'AMA1\_haplotypes\_mutations.txt'

report\_name <- toString(sub(".txt", ".pdf", Protein))

grid <- grid.arrange(bb, nrow=1, ncol=1)

ggsave(grid, file=report\_name, height = 30 , width = 30, device = pdf, dpi = 300)