Akaash Sidhu, 0850326
Bram Ratz, 1068545
Muskan Aggarwal, 1069968
BINF*6970
Friday April 5th, 2019

Assignment 4: Analysis of Genotype Data Available from the 1000 Genomes Project

An exploratory analysis of the OCA2 gene that causes Oculocutaneous albinism (OCA) among 5 super populations

Background

Albinism is a group of disorders characterized by the congenital absence or reduction in melanin [1]. Melanin production is well regulated in the body and synthesized in specialized ectodermal derived cells, known as melanocytes [2]. Oculocutaneous albinism (OCA) is an autosomal recessive disorder that is the result of a disruption in melanin biosynthesis in melanocytes [3]. This form of albinism involves the eyes, skin, and sometimes (depending on the definition), hair [2][3]. There are four types of oculocutaneous albinism, however, for the purpose of this assignment we will be focusing on just one [1]. Oculocutaneous albinism type II (OCA2) is the most common type of albinism, especially among Africans and African Americans [4]. There has notably been a high frequency of OCA2 among specific African populations, such as among the Bamileke people of Cameroon where OCA2 is estimated to be 1 in 7900 and among the Ibo people of Nigeria where it is estimated to be 1 in 1100 [4]. Among African Americans, the frequency of OCA2 is estimated to be 1 in 10000 which is a significant difference when compared to Caucasian Americans, where OCA2 is estimated to affect 1 in 36000 [4]. It is inferred that the higher frequency of OCA2 among African Americans is a reflection of the enslaved African population in the United States [4]. OCA2 is caused by mutations in the P gene and although the specific function of the P gene is unknown it is hypothesized to be involved in tyrosinase processing and transport; tyrosinase is a rate-limiting enzyme in melanin production, a mutation in the P gene results in OCA2 [5][6].

Objective

The purpose of this analysis of OCA2 is to explore if the variation caused by this disorder among populations holds true to the observed higher frequency of OCA2 demonstrated among African and African American populations. This exploratory analysis uses PCA, DPCA, and clustering as a method to classify ethnic populations based on variation among their genetic information.

Description of data

Structural variant data for the OCA2 gene was obtained using data slicer within the 1000 Genomes Project. The OCA2 gene in humans is present on chromosome 15 at 27719008-28099342 bp, which was inputted into data slicer. The data consisted of structural variants of the OCA2 gene from 5 randomly chosen populations from each of the 5 super-populations available, for a total of 25 individual populations. The 5 major super-populations were Africa, the Americas,

Europe, South Asia, and East Asia. Corresponding population information data was downloaded from 1000 Genomes and filtered by sample ID to match the structural variant data.

Data Analysis Methods

In order to analyze the OCA2 gene, the main statistical methods selected were PCA, DPCA, and k-means clustering analysis. For these analyses, the data was filtered such that the minor allele frequency is equal to or greater than 0.001. The data was then indexed to include 200 random variants to decrease computational burden and improve computational speed. Principal component analysis (PCA) was used as it is often used to discover and display patterns in SNP data from humans [7]. It is especially common when trying to explain population structure. K-means clustering analysis was used to explore variation among the ethnic populations. DPCA is performed which is a multivariate statistical approach that uses populations and provide Linkage disequilibrium (LD) as output as we get PCA components in PCA analysis which might be more helpful to maximizes the discrimination between groups.

Results

Figure 1 visualizes a network that represents the minimum distance between each sample from each of the 25 populations. It is apparent based on the sample distribution that the populations are not fully differentiated geographically. There are two groups formed in the data but the effect on segregation is negligible.

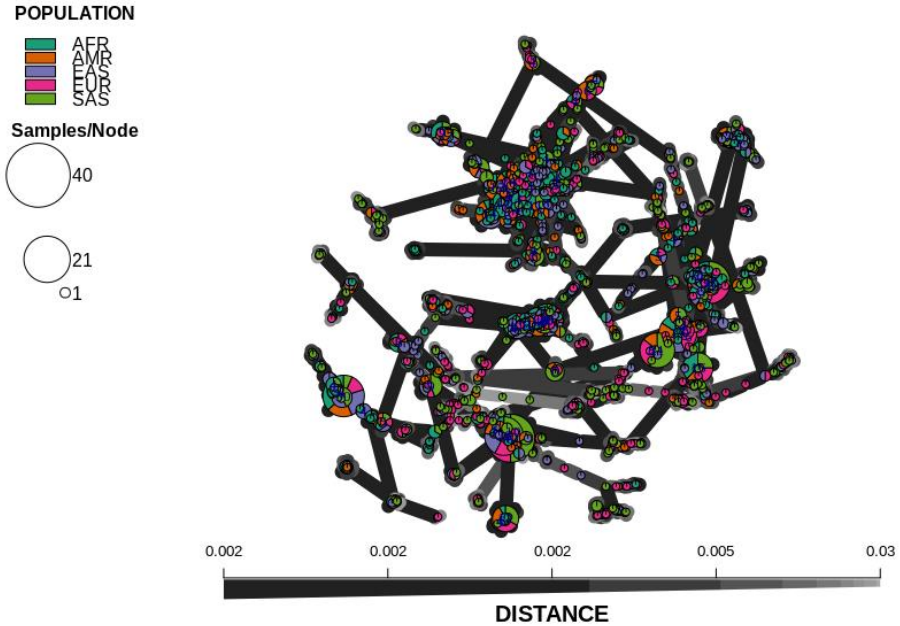

Figure 1: This network represents the minimum distance between each sample from each of the 25 populations. The larger nodes indicated a larger number of samples and the samples are identified based on their associated super population. As described by the bar, the darker the lines are the closer the samples are.

The two principal components were graphed, and samples coloured by the super-population each belong to, shown in Figure 2. It depicts the PCA scores for PCA1 versus PCA2 for the samples in the dataset for the given populations that can describe the pattern based on population however, as illustrated, there is no significant pattern and cannot be defined based on population.

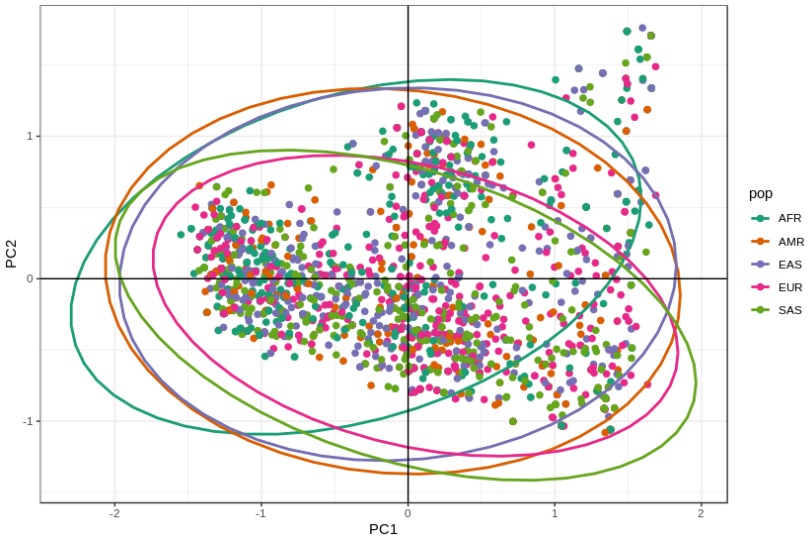

Figure 2: In this visualization of the first two principal components, there is a pattern distinguishable from Figure 1, however, according to the ellipses this pattern is not associated with geographical distribution.

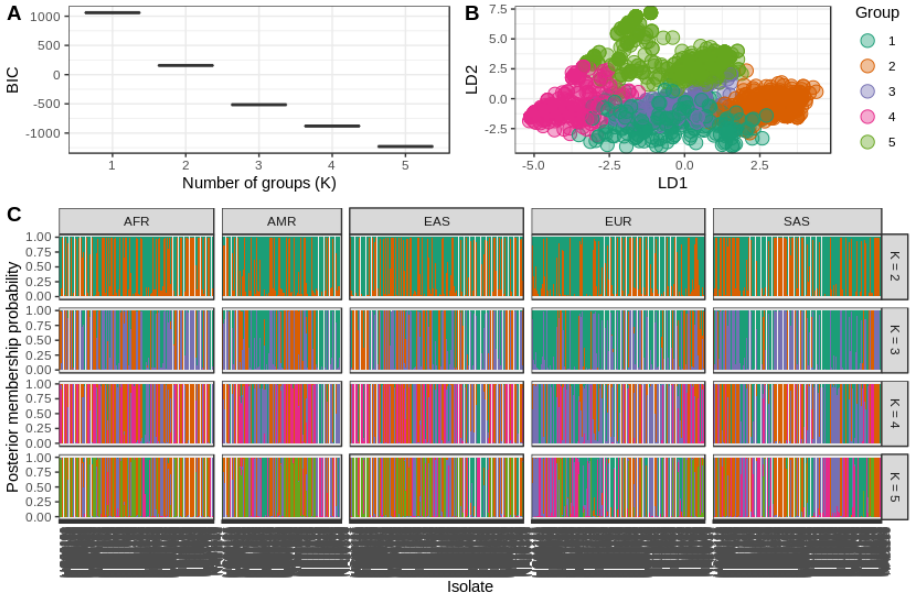

Figure 3: In A, the plot depicts the BIC values which were used to determine the *k* where the lowest BIC value was assigned to 5 groups. In B, there are 5 groups (as indicated by *k*) and 3 distinguishable clusters. It is notable that Group 1 is overlapping Group 3 while the remaining groups create distinctive clusters. In C, it seems that when *k* =5 the variation is spread evenly among the 5 super populations.

In order determine the number of groups needed to conduct the k-means clustering analysis, the BIC values were used to select $k = 5$, figure 3A. K-means clustering grouped the data into 5 groups. As shown in Figure 3B, groups 2, 4, and 5 form distinct groups. Group 3 forms a distinct cluster but is overlapped significantly by group 1. Figure 3C shows the probability of each sample belonging to one of the five super-populations, where the colour represents the group they were clustered to, at different levels of K. The multicoloured distribution seen at K=5 reflects a lack of a samples geographic location to the clustering.

Discussion

According to the research conducted on OCA2, the frequency of OCA2 among African and African American populations should be distinctively different from other populations as it is estimated to have a higher frequency. However, based on the analyses conducted, the super-populations and their subsequent populations, all showed evidence of variation thus distinctive clusters couldn't be made.

The PCA, as illustrated in Figure 2, demonstrates that the PCA1 and PCA2 scores are similarly distributed in all super-populations. So, either of the components cannot be determined on a single super-population.

The clustering seen in Figure 3B, is based on LD, so individuals with closely related SNPs are clustered. We expected that within populations the SNPs would be more closely related as they are a cohesive population in an enclosed geographic area. However, the opposite occurred as geographic location does not dictate how closely related SNPs will be in a specific population and this can be visualized in Figure 3C.

Conclusion

According to our exploratory analysis, individuals with variants in the OCA2 gene cluster based on how closely related their SNPs are, rather than belonging to similar populations. It seems apparent that geographic location plays no role in the frequency or distribution of SNPs among individuals in the OAC2 gene.

In the future, research done for OCA2 should focus on classifying based on hereditary information instead of a population genetics approach [8]. This can be done by using pedigree data instead of population distribution data obtained from 1000 Genome to conduct statistical analyses.

Citations

[1] Rees, J. L. (2003). Genetics of hair and skin color. *Annual review of genetics*, *37*(1), 67-90.

[2] Kamaraj, B., & Purohit, R. (2014). Mutational analysis of oculocutaneous albinism: a compact review. *BioMed research international*, *2014*.

[3] Kamaraj, B., & Purohit, R. (2014). Computational screening of disease-associated mutations in OCA2 gene. *Cell biochemistry and biophysics*, *68*(1), 97-109.

[4] Puri, N., Durham-Pierre, D., Aquaron, R., Lund, P. M., King, R. A., & Brilliant, M. H. (1997). Type 2 oculocutaneous albinism (OCA2) in Zimbabwe and Cameroon: distribution of the 2.7-kb deletion allele of the P gene. *Human genetics*, *100*(5-6), 651-656.

[5] Oetting, W. S., Garrett, S. S., Brott, M., & King, R. A. (2005). P gene mutations associated with oculocutaneous albinism type II (OCA2). *Human mutation*, *25*(3), 323-323.

[6] Rimoldi, V., Straniero, L., Asselta, R., Mauri, L., Manfredini, E., Penco, S., ... & Primignani, P. (2014). Functional characterization of two novel splicing mutations in the OCA2 gene associated with oculocutaneous albinism type II. *Gene*, *537*(1), 79-84.

[7] Gauch, H. G., Qian, S., Piepho, H. P., Zhou, L., & Chen, R. (2018). Effective principal components analysis of SNP data. *bioRxiv*, 393611.

[8] Hu, H., Wang, H., Jia, Z., & Xie, Q. (2014). Prenatal genetic diagnosis for two Chinese families affected with oculocutaneous albinism type II. *Zhonghua yi xue yi chuan xue za zhi= Zhonghua yixue yichuanxue zazhi= Chinese journal of medical genetics*, *31*(4), 424-427.

Appendix

```
###-------------------------------------------------------------

### An exploratory analysis of the OCA2 gene that causes Oculocutaneous
albinism (OCA) among 5 super populations
### BINF6970 Assignment 4
### Group # 6
### Group members:
      #Akaash Sidhu: 0850326
      #Bram Ratz: 1068545
      #Muskan Aggarwal: 1069968

###-------------------------------------------------------------

## Required packages
library(tidyverse)
library(vcfR)
library(poppr)
library(ape)
library(RColorBrewer)
library(igraph)
library(ggplot2)
library(adegenet)
library(adegraphics)
library(trio)
library(lattice)
library(ape)
library(reshape2)
library(snpStats)
library(ggpubr)

## set working directry
#setwd("~/Desktop/BINF6970/Homework_assign4")

## load VCF file for OAC1 into R
vcf_OCA2 <- read.vcfR("15.27965575-
28378929.ALL.chr15.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes-
1.vcf.gz")

## see whats actually in the file
#summary of vcf file
vcf_OCA2

#can see from this we have:
#2292 samples
#0% missing

#use head to view samll section of the file
head(vcf_OCA2)

#look at the fix region - contains info about each variantin the sample
head(getFIX(vcf_OCA2))

#look at the gt region - contains info genotype info about each variant in
the sample
```

```
vcf_OCA2@gt[1:6, 1:4]

## use chromR to locate unusual features in a genome
#make chrom object
chrom <- create.chromR(name = 'OCA2_data', vcf = vcf_OCA2)

#plot the chrom object
plot(chrom)

###-------------------------------------------------------
### subset population to speed computation time
###-------------------------------------------------------

## Remove rare alleles across the pop - minor allele frequency < 0.001
vcf_maf <- maf(vcf_OCA2, element = 2)

#turn into a df
vcf_maf_df <- as.data.frame(vcf_maf)

#subset on basis of frequency less than 0.001
vcf_maf_sub <- vcf_maf_df[vcf_maf_df$Frequency >= 0.001,]

#compare counts
count(vcf_maf_df)
count(vcf_maf_sub)

#remove rsIDs
vcf_rsID <- vcf_OCA2@fix[,3]

#index to column of the vcf_maf_sub object
vcf_maf_sub$names <- rownames(vcf_maf_sub)
head(vcf_maf_sub)

#make vector for the rsID
names <- vcf_maf_sub$names

#filter
vcf_filter <- which(names %in% vcf_rsID)

vcf_OCA2 <- vcf_OCA2[vcf_filter,]

#make sure this works
class(vcf_OCA2)
head(vcf_OCA2)
vcf_OCA2

## want to maintain the same number of samples but get a subset of 200 random
variants to make it easier to work with
#get vector of 200 random variants from vcf file
subset1 <- sample(size = 200, x = c(1:nrow(vcf_OCA2)))

#use vector of 200 variants to subset the vcf file into a more managable size
vcf_sub_OCA2 <- vcf_OCA2[subset1,]

#confirm we have the same number of samples but only 200 variants
vcf_sub_OCA2
```

```
###-------------------------------------------------------
### converting the dataset to a genlight object
###-------------------------------------------------------
## convert to genlight
#OCA2.genlight <- vcfR2genlight(vcf_OCA2, n.cores = 1)
OCA2.genlight <- vcfR2genlight(vcf_sub_OCA2, n.cores = 1)

#conversion removes 20 loci that have more than two alleles

#make sure to specifiy a ploidy of 2
ploidy(OCA2.genlight) <- 2

## using which to find the samples that are the same across two dataframes
#get excel data
excel.data <- read.csv("Sample-info.csv")

#subset
excel.sub <- as.character(excel.data[,1])
names.sub <- indNames(OCA2.genlight)

#compare the two
SameSamples <- which(excel.sub %in% names.sub)

excel.final <- excel.data[SameSamples,]

#check to make sure worked
head(excel.final)

#adding real snp names
locNames(OCA2.genlight) <- paste(vcf_OCA2@fix[,1], vcf_OCA2@fix[,2], sep =
"_")

#add population names
pop(OCA2.genlight) <- excel.final$Superpopulation.code

## check the object to make sure this actually worked
#basic info of the object
OCA2.genlight

#checking individual names
indNames(OCA2.genlight)

#look at small amount of data
as.matrix(OCA2.genlight)[1:16, 1:10]

#look at the populations associated with the sample
pop(OCA2.genlight)

## create variable to colour to use for the rest of the script
n <- 60
qual_col_pals = brewer.pal.info[brewer.pal.info$category == 'qual',]
col_vector = unlist(mapply(brewer.pal, qual_col_pals$maxcolors,
rownames(qual_col_pals)))

cols <- brewer.pal(n = nPop(OCA2.genlight), name = "Dark2")

###-------------------------------------------------------
```

```
### Minimum spanning networks
###-------------------------------------------------------

## Make minimum spanning network to visualize patterns in the data
## Need a genlight object and distance matrix to construct a MSN
#make a distance matrix using bitwise.dist
OCA2.dist <- bitwise.dist(OCA2.genlight)

#make MSN
OCA2.msn <- poppr.msn(OCA2.genlight,
                      OCA2.dist,
                      showplot = FALSE,
                      include.ties = T)

#attibute variables
node.size <- rep(2, times = nInd(OCA2.genlight))
names(node.size) <- indNames(OCA2.genlight)
vertex.attributes(OCA2.msn$graph)$size <- node.size

#set seed
set.seed(4419)

#plot the spanning network
plot_poppr_msn(OCA2.genlight,
               OCA2.msn,
               palette = brewer.pal(n = nPop(OCA2.genlight),
                                    name = "Dark2"),
               gadj = 70,
               inds = "one")

###-------------------------------------------------------
### PCA
###-------------------------------------------------------

##PCA analysis to look for sources of variation between samples

OCA2.pca <- glPca(OCA2.genlight, nf = 5)

#make a bar plot to show the percentage of variation each PC explains
barplot(100*OCA2.pca$eig/sum(OCA2.pca$eig),
        col = heat.colors(50),
        main = "PCA Eigenvectors")

#add labels
title(ylab = "Percentage of variance\nexplained", line = 2)
title(xlab = "Eigenvalues", line = 1)

#can see from this that the first 5 PCA seem to be the one needed

## proportion of variance explained by the first three axes
#first axis
OCA2.pca$eig[1]/sum(OCA2.pca$eig)

#second axis
OCA2.pca$eig[2]/sum(OCA2.pca$eig)

#third axis
```

```
OCA2.pca$eig[3]/sum(OCA2.pca$eig)

sum(OCA2.pca$eig[1:5]/sum(OCA2.pca$eig))
#first 3 PCA's explain 72% of the variance

#how many PCs to keep
plot(as.ts(OCA2.pca$eig[1:10]),
     ylab = "Eigenvalues",
     xlab = "components",
     main = "Scree plot")

#want first 5 PCs

## can view the results of the PCA graphically using ggplot2
#make df for the pca scores
OCA2.pca.scores <- as.data.frame(OCA2.pca$scores)
OCA2.pca.scores$pop <- pop(OCA2.genlight)

#set seed again
set.seed(4419)

#plot the results of the pca
pca.plot <- ggplot(OCA2.pca.scores,
                   aes(x = PC1, y = PC2, colour = pop))
pca.plot <- pca.plot + geom_point(size = 2)
pca.plot <- pca.plot + stat_ellipse(level = 0.95, size = 1)
pca.plot <- pca.plot + scale_color_manual(values = cols)
pca.plot <- pca.plot + geom_hline(yintercept = 0)
pca.plot <- pca.plot + geom_vline(xintercept = 0)
pca.plot <- pca.plot + theme_bw()

pca.plot

###-------------------------------------------------------
### DAPC
###-------------------------------------------------------

## Use to maximize the discrimination between groups
#use same results as pca to make the results compareable

## make DAPC object
#using n.pca = 3 b/c retained 3 PC for the pca
OCA2.dapc <- dapc(OCA2.genlight, n.pca = 3, n.da = 2)

#confirm that DAPC is similar to PCA by plotting
scatter(OCA2.dapc,
        col = cols,
        cex = 2,
        legend = TRUE,
        clabel = F,
        posi.leg = "bottomleft",
        scree.pca = TRUE,
        posi.pca = "topleft",
        cleg = 0.75)

#the results of the DAPC and PCA are very similar
```

```
## use ggplot2 to construct plot
#extract DAPC calculated pop membership assignment into new df
dapc.results <- as.data.frame(OCA2.dapc$posterior)

#include original pop assignment
dapc.results$pop <- pop(OCA2.genlight)

#add a column that includes sample names
dapc.results$indNames <- rownames(dapc.results)

#transform df using melt into format we need
dapc.results <- melt(dapc.results)

#rename columns
colnames(dapc.results) <- c("original_pop", "samples", "assigned_pop",
"posterior_membership_probability")

#ggplot2 will plot dapc.results df using samples on the x-axis and membership
probabilities on the Y-axis. fill colour will indicate orignial pop
assignment
dapc.plot <- ggplot(dapc.results,
                    aes(x=samples,
                        y=posterior_membership_probability,
                        fill=assigned_pop))
dapc.plot <- dapc.plot + geom_bar(stat='identity')
dapc.plot <- dapc.plot + scale_fill_manual(values = cols)
dapc.plot <- dapc.plot + facet_grid(~original_pop, scales = "free")
dapc.plot <- dapc.plot + theme(axis.text.x = element_text(angle = 90, hjust =
1, size = 8))
dapc.plot

#the bar plot shows more organized perspective by contrasting the pop
membership probability assignments against their original population
#shows that every region shows a mix of of all other regions
  #share similar variants acorss the all geopgraphic regions

#***************************************************************
### K-means
#***************************************************************

## Use a BIC graph to determine optimium number of clusters

maxK <- 5
#Set maximum number to 5

myMat <- matrix(nrow=5, ncol=maxK)
#Create a 5 by 5 matrix

colnames(myMat) <- 1:ncol(myMat)
#Set column names

for(i in 1:nrow(myMat)){
  grp <- find.clusters(OCA2.genlight, n.pca = 5, choose.n.clust = FALSE,
max.n.clust = maxK)
  myMat[i,] <- grp$Kstat
}
#Created for loop to assign data to clusters sequenctially
```

```
my_df <- melt(myMat)
#In order to visualize the clusters, a new dataframe is created.

colnames(my_df)[1:3] <- c("Group", "K", "BIC")
my_df$K <- as.factor(my_df$K)
#Set column names

head(my_df)
#View the new dataframe

p1 <- ggplot(my_df, aes(x = K, y = BIC))
p1 <- p1 + geom_boxplot()
p1 <- p1 + theme_bw()
p1 <- p1 + xlab("Number of groups (K)")
p1
#Plot the K against BIC

##FOR: DAPC
#set the number of groups to between 2 to 5
my_k <- 2:5

grp_l <- vector(mode = "list", length = length(my_k))
dapc_l <- vector(mode = "list", length = length(my_k))

for(i in 1:length(dapc_l)){
  set.seed(9)
  grp_l[[i]] <- find.clusters(OCA2.genlight, n.pca = 5, n.clust = my_k[i])
  dapc_l[[i]] <- dapc(OCA2.genlight, pop = grp_l[[i]]$grp, n.pca = 5, n.da =
my_k[i])
}
#takes a long time to create the clusters

my_df <- as.data.frame(dapc_l[[ length(dapc_l) ]]$ind.coord)
my_df$Group <- dapc_l[[ length(dapc_l) ]]$grp
head(my_df)
#Viewing differentiation of our data into different clusters

my_pal <- RColorBrewer::brewer.pal(n=8, name = "Dark2")

p2 <- ggplot(my_df, aes(x = LD1, y = LD2, color = Group, fill = Group))
p2 <- p2 + geom_point(size = 4, shape = 21)
p2 <- p2 + theme_bw()
p2 <- p2 + scale_color_manual(values=c(my_pal))
p2 <- p2 + scale_fill_manual(values=c(paste(my_pal, "66", sep = "")))
p2
#Plot the data

tmp <- as.data.frame(dapc_l[[1]]$posterior)
tmp$K <- my_k[1]
tmp$Isolate <- rownames(tmp)
tmp <- melt(tmp, id = c("Isolate", "K"))
names(tmp)[3:4] <- c("Group", "Posterior")
tmp$Region <- excel.final$Superpopulation.code
my_df <- tmp
#View as a barplot, use facets to separate the different values of K. Combine
data into single long form data.frame and add population data
```

```
for(i in 2:length(dapc_l)){
  tmp <- as.data.frame(dapc_l[[i]]$posterior)
  tmp$K <- my_k[i]
  tmp$Isolate <- rownames(tmp)
  tmp <- melt(tmp, id = c("Isolate", "K"))
  names(tmp)[3:4] <- c("Group", "Posterior")
  tmp$Region <- excel.final$Superpopulation.code

  my_df <- rbind(my_df, tmp)
}

grp.labs <- paste("K =", my_k)
names(grp.labs) <- my_k

p3 <- ggplot(my_df, aes(x = Isolate, y = Posterior, fill = Group))
p3 <- p3 + geom_bar(stat = "identity")
p3 <- p3 + facet_grid(K ~ Region, scales = "free_x", space = "free",
                      labeller = labeller(K = grp.labs))
p3 <- p3 + theme_bw()
p3 <- p3 + ylab("Posterior membership probability")
p3 <- p3 + theme(legend.position='none')
p3 <- p3 + scale_fill_manual(values=c(my_pal))
p3 <- p3 + theme(axis.text.x = element_text(angle = 90, hjust = 1, size = 8))
p3
#Build the ggplot

ggarrange(ggarrange(p1,p2, ncol = 2, labels = c("A", "B")), p3, nrow = 2,
labels = c("", "C"), heights = c(1, 2))
#Put this all together into one plot using ggpubr
```