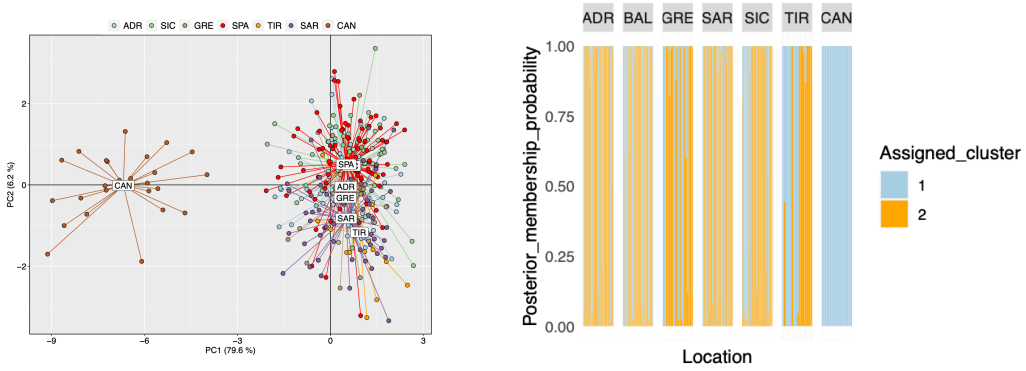
Dear authors,

I have received your resubmission, and on the overall, you addressed the reviewers' comments passably. However, some points remained inadequately dealt with. My opinion is that before sending the manuscript out for another round of reviews, I would like you to take my comments on board and make whichever changes you see fit (please check the scripts at the end of this file). I think you could have invested a bit more time to illustrate your results and claims so that reviewers and hopefully the readers can fully appreciate your paper. Overall, I maintain that the MS and the data have the potential to be an important contribution to the understanding of population structuring in an economically important fishery species. However, as I said, the MS still needs improvements, and thus I recommend major revisions again.
I look forward to seeing a revised version of this MS in the near future.

General remarks.

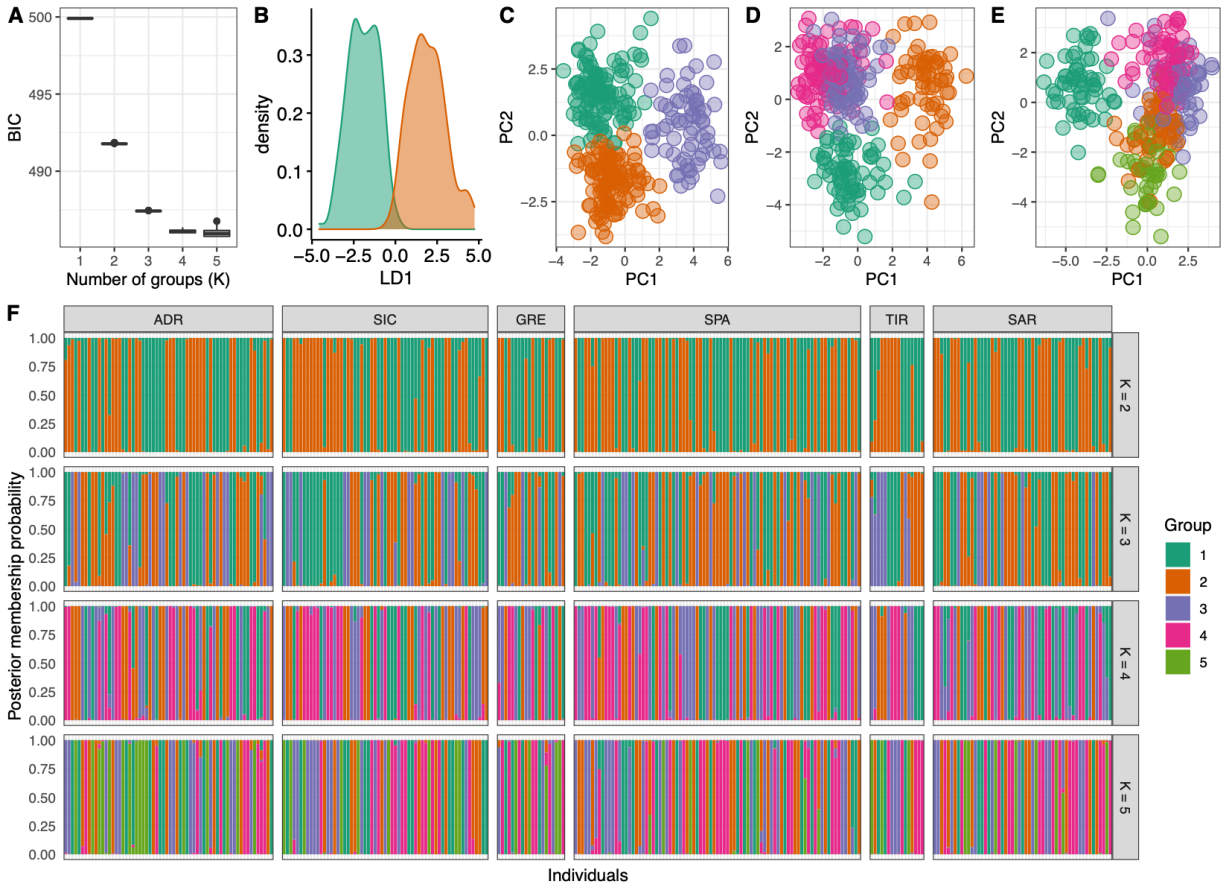
I agree with the reviewers that at first glance your results seem to indicate two clades in the Mediterranean and not three clades as you argue. However, I have run your data and using the *find.clusters* function [find.clusters(med.data, n.pca=100, choose=FALSE, max.n.clust = 6)] I have found indeed that it is also possible the existence of three groups in the Mediterranean. Because you have not shared your script, I cannot check how you have run the DAPC. However, the DAPC I have run does not show such a clear difference among the Mediterranean locations as the one you found. Therefore I would present a figure of the DAPC with all locations, as you did (I am sending you the script - *0_DAPC_swordfish_peerj.R* - for the figure in case you want to use them



or just keep the one you produced).

Then run a DAPC only for the Mediterranean samples. From my viewpoint, I would not even use STRUCTURE, and therefore, I would suggest you replace Figures 2 and 3 by a unique larger figure where you display the deltaK, plots for K= 2, 3, 4 and 5. I am providing the script to produce this figure too - *1_swordfish_med.R* - (please acknowledge Tom Jenkins (https://github.com/Tom-Jenkins) and BJ Knaus, and NJ Grünwald (https://grunwaldlab.github.io/Population_Genetics_in_R/clustering_plot.html).

Given these results, I think you should make room in the text for the two possibilities: "In the DAPC clustering analysis, we selected K = 2 and K = 3 structures based on the BIC curve which represents the plausible number of clusters in the data."



Minor suggestions.
Figure 1. Please add codes of Table 1. Otherwise, the reader cannot locate sampling sites. Table 3 and 4 to be removed. Results can be referred to in the text.

Line 26. "… and their conclusions are rather controversial and not yet conclusive" conveys the same idea. Replace by "and conclusions are controversial".

Line 27. Include the number of microsatellite loci used.

Line 29. Start by stating your main results: "This study provides evidence to reject the hypothesis of a single swordfish population within the Mediterranean Sea. DAPC analysis revealed the presence of two or three genetic clusters and a high level of admixture within the Mediterranean Sea. Genetic structure was supported by significant FST values while mixing was endorsed by the heterozygosity deficit observed in sampling localities indicative of a possible Wahlund effect, by sampling admixture individuals.

Line 29. Remove "In this study, no evidence of differentiation was detected between sampling areas."

Line 38. Introduction. You should mention ATL studies as well

Line 41. "Fish species frequently form different populations, which are genetically structured through behavioural and distributional differences (Reiss et al., 2009)." Add "geographical" before distributional.

Line 45. Remove i) and ii)

Line 70. Replace "biological" by "biology" and delete "terms"

Line 71. Delete "biological"

Line 74. Replace "these" by "the Atlantic and Mediterranean"

Line 80. Remove "were detected among these basins"

Line 87. Before the end of the sentence add (Figure 1)

Lines 94-100. Replace by "While there are several studies on the global genetic structure of swordfish, few have focused on the Mediterranean swordfish stock. Some of these studies suggest the occurrence of a homogeneous stock in the Mediterranean Sea (RFLP of the entire mitochondrial DNA, Kotoulas et al., 1995; RFLP of the mitochondrial control region and the nuclear Calmodulin gene, Chow and Takeyama, 2000; four microsatellites, Kotoulas et al., 2007).

Line 108. Remove "more"

Line 109. Remove "more"

Line 110. State how many microsatellites were used.

Line 119. Please do not start a sentence with a numeral

Line 197. I think the sentence "Pairwise F_{ST} values among genetic clusters were calculated using Arlequin." is misplaced here.

Lines 197-205. I think this text is also misplaced here, it should go inline 176, before "The Bayesian"

Line 236. Fst values range from 0 to 1. Therefore negative values should be changed to zero

Line 238. After Atlantic samples include ", Fst values ranging from 0.083 (CAN-GRE) to 0.097 (CAN-SAR).

Line 243. The assignment of individuals is not mentioned in the Material and Methods section

Line 275. Before entering the discussion in detail, you should refer to the main caveat of the study: sampling sites. I suggest something like: "Before examining the results, we must address the mains caveat of this study: sampling sites. The samples for this study were collected opportunistically. Therefore we cannot evaluate hypothesis related to putative homing of the species towards breeding areas or to evaluate admixture in the feeding locations."

Line 288. Replace (around 0.09) by the exact value

Line 289. Add "also" after works

Line 300. Modify to "concordant, with DAPC providing evidence"

Line 305. Delete the "s" from "fails"

Lines 362-368. This text seems misplaced here. It pertains to the introduction

Line 371. Replace "draw definite conclusions" by "test this hypothesis"

Line 382. After "The present study suggests genetic heterogeneity within the Mediterranean Sea swordfish stock" Add "supporting previous studies (REF, REF, REF).

Line 386. Delete "and philopatry"

Line 397. Remove "population in the"
END ———————————————————————————————————————-

0_DAPC_all_swordfish_peerj.R

```
#' ---
#' title: "DAPC swordfish microsatellite analysis"
#' author: "Rita Castilho rcastil@ualg.pt"
#' date: "APR 2020"
#' ---

# install and load packages
# set working directories
# load datafile
# DAPC using all micros WITH geographical information
# Visualise DAPC using all microsatellites
# Posterior membership probabilities "individuals to pop"
#    to generate plots similar to those generated for STRUCTURE
# DAPC using ONLY Mediterranean data
# MED DAPC using all micros WITH geographical information
# visualise MED DAPC using all microsatellites
# Mediterranean Posterior membership probabilities "individuals to pop"
#     to generate plots similar to those generated for STRUCTURE
# combine plots the using cowplot package

#-------------------------------------------------------------------------------#
# install and load packages
#-------------------------------------------------------------------------------
check.packages <- function(pkg){
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
  if (length(new.pkg))
    install.packages(new.pkg, dependencies = TRUE)
  sapply(pkg, require, character.only = TRUE)
}

packages<-c("adegenet","reshape","ggplot2","cowplot",
            "snow","poppr","cowplot", "devtools")
check.packages(packages)

install_github("romunov/zvau")
library(zvau)

#-------------------------------------------------------------------------------
# set working directories
#-------------------------------------------------------------------------------
# what is the name of the main root folder?
# remember that you have created the TP_02 in the last script
root <- "~/Desktop"
dir <- "PeerJ_swordfish"
dir.create(paste(root,dir,sep="/"))
root <- paste(root,dir,sep="/")
setwd(root)
getwd()

#-----------------------------------------------------------
# where to place results directory
#-----------------------------------------------------------#
folder.results <- "r.results"
dir.create(paste(folder.results,sep="/"))
folder.results <- folder.results
#-----------------------------------------------------------#
# where is code?
```

```
#-----------------------------------------------------------#
folder.code <- "r.code"
dir.create(paste(folder.code,sep="/"))
folder.code <- paste(root,folder.code,sep="/")

#-----------------------------------------------------------#
# where to retrieve data files from?
#-----------------------------------------------------------#
folder.data <- "r.data"
dir.create(paste(root,folder.data,sep="/"))
folder.data <- paste(root,folder.data,sep="/")

# option 1
data.file <- "swordfish.gen"
data <- read.genepop('http://rcastilho.pt/MARGEN/ewExternalFiles/swordfish.gen',ncode=3)

#-------------------------------------------------------------------------------#
# create list of population names
#-------------------------------------------------------------------------------#
levels(data@pop) <-c(
  "ADR", "BAL", "GRE", "SAR", "SIC", "TIR","CAN"
)

(lab <- levels(data@pop))
(k <- length(lab))

## colour definitions
cols <- c("#A6CEE3","orange","#99CD91","#B89B74","red","#825D99","#B15928")

#-------------------------------------------------------------------------------
# 2
# find.clusters using all micros NO apriori geographical information
#-------------------------------------------------------------------------------
## run the DAPC with no apriori of location with CANADA

clusters <- find.clusters(data, n.pca=100, choose=FALSE, max.n.clust = 6)
#-------------------------------------------------------------------------------#
# find.cluster visualization
table(clusters$grp)
table(pop(data), clusters$grp)
table.value(
  table(pop(data),
        clusters$grp),
  col.lab=paste("cluster", 1:6),
  row.lab=paste(c(lab)),
  grid = FALSE,
  clegend = 0
)
dev.copy(pdf,paste(folder.results,"cluster_grid_with Canada.pdf",sep="/"),width=10,
height=8)
def.off()

#-------------------------------------------------------------------------------
# 3.
# run the DAPC with groups from find.clusters (without apriori location)
# plot DAPC
#-------------------------------------------------------------------------------
dapc.priori <- dapc(data, clusters$grp,n.pca=100, choose=FALSE, n.da=2) # a priori dapc
scatter(dapc.priori,2,2,col=cols,scree.da=F,legend=T,solid=0.6) # plot
data.noNA<-tab(data, NA.method="mean") #if there are NA, replace
pca_res.priori <- dudi.pca(data.noNA, scannf=FALSE, nf = 2) # pca

#-------------------------------------------------------------------------------#
# convert pca, kmeans and original pops to a dataframe
points.priori <- data.frame(x = pca_res.priori$li[,1],
                            y = pca_res.priori$li[,2],
                            inferred = clusters$grp,
                            original = data@pop
)

#-------------------------------------------------------------------------------
# 4
# posterior membership probabilities "individuals to cluster"
```

```
# to generate plots similar to those generated for STRUCTURE
#-------------------------------------------------------------------------
dat <- melt(dapc.priori$posterior)

# to visualize it better, add the pop information through ggplot2:
dapc.posteriori <- as.data.frame(dapc.priori$posterior)
dapc.posteriori$pop <- pop(data)
dapc.posteriori$indNames <- rownames(dapc.posteriori)
dapc.posteriori <- melt(dapc.posteriori)
colnames(dapc.posteriori) <-
c("Original_Pop","Location","Assigned_cluster","Posterior_membership_probability")

#-------------------------------------------------------------------------#
(p1 <- ggplot(dapc.posteriori, aes(x=Location, y=Posterior_membership_probability,
fill=Assigned_cluster))+
    geom_bar(stat='identity')+
    scale_fill_manual(values = cols)+
    facet_grid(~Original_Pop, scales = "free")+
    theme(axis.text.x = element_blank(),axis.ticks = element_blank())
)

ggsave(paste(folder.results,"structure-like_can.pdf",sep="/"),
       plot = p1,
       device = "pdf",
       scale = 1,
       width = 11,
       height = 8,
       units = "cm"
)
#-------------------------------------------------------------------------#


#-------------------------------------------------------------------------
# 5
# DAPC using all micros WITH geographical information
#-------------------------------------------------------------------------
## cross validation to find the optimal number of PCs to retain in DAPC

x.all <- tab(data, NA.method="mean")
val.all <- xvalDapc(x.all, data$pop, n.pca.max=100, training.set=0.9,
           result="groupMean", center=TRUE, scale=FALSE,
           n.rep=10, n.pca=NULL)
## number of PCs with best stats
nPC <- as.numeric(val.all$"Number of PCs Achieving Highest Mean Success")
val.all$`Number of PCs Achieving Lowest MSE`
val.all$`Root Mean Squared Error by Number of PCs of PCA` # lower score = better

## run the DAPC using population IDs as priors
dapc.all <- dapc(data, data$pop, n.pca=nPC, n.da=3)

## analyse how much percent of genetic variance is explained by each axis
(percent <- dapc.all$eig/sum(dapc.all$eig)*100)
barplot(percent,
        names.arg=round(percent, 2))
mtext(side = 1, text = "PC", line = 4)
mtext(side = 2, text = "Percent of genetic variance explained \nby eigenvectors", line =
2)

dev.copy(pdf,paste(folder.results,"barplot.pdf", sep="/"),width=10, height=8)
def.off()

## microsatellite contributions
contrib.all <- (dapc.all$var.contr)
#contrib.all <- loadingplot(dapc.all$var.contr, threshold=0.005)
#contrib.all$var.names
## Create data_filtframe with PC info
df_pca.all <- as.data.frame(dapc.all$ind.coord)
## Add a column containing individuals
df_pca.all <- cbind(rownames(df_pca.all), df_pca.all)
## Rename first three columns
colnames(df_pca.all) <- c("Indiv","PC1","PC2","PC3")
## Add a column with the population IDs
df_pca.all$pop <- data$pop
```

```
## Flip axis by multiplying by minus 1
df_pca.all[ , 2:4] <- df_pca.all[ , 2:4]*-1

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

#-------------------------------------------------------------------------
# Visualise DAPC using all microsatellites
#-------------------------------------------------------------------------
## ggplot2 theme
ggtheme <- theme(legend.title = element_blank(),
                 axis.text.y = element_text(colour="black", size=14),
                 axis.text.x = element_text(colour="black", size=14),
                 axis.title = element_text(colour="black", size=14),
                 legend.position = "top",
                 legend.text = element_text(size=15),
                 legend.key.size = unit(0.7,"cm"),
                 legend.box.spacing = unit(0, "cm"),
                 panel.border = element_rect(colour="black", fill=NA, size=1),
                 plot.title = element_text(hjust=0.5, size=25) # title centered
)

## calculate centroid position for each population
centroid.all <- aggregate(cbind(PC1, PC2, PC3) ~ pop, data=df_pca.all, FUN=mean)

## find and store coordinate info required to draw segments
segs.all <- merge(df_pca.all, setNames(centroid.all,
c("pop","oPC1S1","oPC2S2","oPC3S3")),
              by = "pop", sort = FALSE)

## definitions
(no_micros.all = nLoc(data))
(no_ind.all = nInd(data))

## scatter plot axis 1 vs 2
(all <- ggplot(df_pca.all, aes(x=PC1, y=PC2))+
       geom_hline(yintercept = 0)+
       geom_vline(xintercept = 0)+
       # spiders
       geom_segment(data=segs.all, mapping = aes(xend=oPC1S1, yend=oPC2S2,
                                           colour=pop), show.legend=FALSE)+
       geom_point(aes(fill=pop), shape=21, size=3, show.legend=TRUE)+
       # ellipses
       #stat_ellipse(level=0.80, size=1)+
       # centroids
       geom_label(data=centroid.all, size=5, aes(label=pop), show.legend=FALSE)+
       scale_fill_manual(values=cols)+
       scale_color_manual(values=cols)+
       labs(x=paste("PC1 (",format(round(percent[1], 1), nsmall=1)," %)", sep=""))+
       labs(y=paste("PC2 (",format(round(percent[2], 1), nsmall=1)," %)", sep=""))+
       panel_border(colour="black", size=0.8)+
       ggtheme+
       guides(fill=guide_legend(nrow=1, byrow=TRUE))
)

## export plot
ggsave(paste(folder.results,"DAPC_all.pdf",sep="/"), width=10, height=8)

#-------------------------------------------------------------------------
# Posterior membership probabilities "individuals to pop"
#    to generate plots similar to those generated for STRUCTURE
#-------------------------------------------------------------------------
dat <- melt(dapc.all$posterior)

# to visualize it better, add the pop information through ggplot2:
dapc.posteriori <- as.data.frame(dapc.all$posterior)
dapc.posteriori$pop <- pop(data)
dapc.posteriori$indNames <- rownames(dapc.posteriori)
dapc.posteriori <- melt(dapc.posteriori)
colnames(dapc.posteriori) <-
c("Original_Pop","Location","Assigned_cluster","Posterior_membership_probability")

# plot
```

```
(p2 <- ggplot(dapc.posteriori, aes(x=Location, y=Posterior_membership_probability,
fill=Assigned_cluster))+
    geom_bar(stat='identity')+
    scale_fill_manual(values = cols)+
    facet_grid(~Original_Pop, scales = "free")+
    theme(axis.text.x = element_blank(),axis.ticks = element_blank())
)

# save
ggsave(paste(folder.results,"structure-like_can_assign.pdf",sep="/"),
       plot = p2,
       device = "pdf",
       scale = 1,
       width = 11,
       height = 8,
       units = "cm"
)


#------------------------------------------------------------------------
# END
#------------------------------------------------------------------------
```

1_DAPC_med_swordfish_peerJ

```
#' ---
#' title: "K-means and DAPC MEDITERRANEAN SWORDFISH"
#' author: "tom Jenkins t.l.jenkins@exeter.ac.uk
#' author: https://grunwaldlab.github.io/Population_Genetics_in_R/clustering_plot.html
#' modified:  Rita Castilho rcastil@ualg.pt"
#' date: "MAY 2020"
#' ---


#------------------------------------------------------------
# 1.preparation
#------------------------------------------------------------

#------------------------------------------------------------
# 1.1 install and load packages
#------------------------------------------------------------
check.packages <- function(pkg){
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
  if (length(new.pkg))
    install.packages(new.pkg, dependencies = TRUE)
  sapply(pkg, require, character.only = TRUE)
}

packages<-c("adegenet","reshape","ggplot2","ggpubr","poppr", "dplyr")
check.packages(packages)


#------------------------------------------------------------
# 1.2 set working directories
#------------------------------------------------------------
# what is the name of the main root folder?
root <- "~/Desktop/PeerJ_swordfish/"
setwd(root)
getwd()

##------------------------------------------------------------
# where to retrieve data files from?
folder.data <- paste(root,"r.data",sep="")
(filelist <- list.files(folder.data, full.names=F))

#------------------------------------------------------------
# where to place results directory
folder.results <- "r.results"
dir.create(paste(root,folder.results,sep=""))
```

```
folder.results <- paste(root,folder.results,sep="")

#-----------------------------------------------------------
# where is code?
folder.code <- "r.code"
folder.code <- paste(root,folder.code,sep="")

#-----------------------------------------------------------
# load datafile
#-----------------------------------------------------------
# filename?
# consider individual SAR22 also removed for being an outlier as in file
swordfish_322.gen
# otherwise use swordfish.gen
data <- read.genepop(paste(folder.data,"swordfish_322.gen",sep="/"),ncode=3)
lab <- levels(data@pop)

#-----------------------------------------------------------
# create list of population names
levels(data@pop) <-c(
  "ADR", "SIC", "GRE", "SPA", "TIR",
  "SAR", "CAN"
)
lab <- levels(data@pop)
k <- length(lab)

cols<-c("#A6CEE3","#99CD91","#B89B74","red","orange","#825D99","#B15928")

#-----------------------------------------------------------
# do you want to remove Canada?
# (un)comment next two lines
data <- popsub(data, sublist=1:6)
data$pop
#-----------------------------------------------------------

# It is recommended to explore several values of K.
# The find.clusters() function includes some stochasticity.
# When we,Äôre at the figure creation step we,Äôll need consistency,
# so we,Äôll set a seed.
# If you,Äôre at an earlier stage in your analysis you should comment
# the set.seed() call out to explore how sensitive your results are to the seed.

# start by performing K-means clustering over a number
# of values of K and repeat 500 times for each value
# so we can explore variability for these values.
maxk <- 5
rep <- 500 #10
mat <- matrix(nrow=rep, ncol=maxk)
colnames(mat) <- 1:ncol(mat)

# K-means clustering - if there is no clear inflection point, retain all PC's
for(i in 1:nrow(mat)){
  grp <- find.clusters(data, n.pca = 100, choose.n.clust = FALSE, max.n.clust = maxk)
  mat[i,] <- grp$Kstat
}

(grp$Kstat)
(grp$grp)

# find.cluster visualization
table(grp$grp)
table(pop(data), grp$grp)
t <- table.value(
  table(pop(data),
        grp$grp),
  col.lab=paste("cluster", 1:6),
  row.lab=paste(c(lab)),
  grid = FALSE,
  clegend = 0
)

dev.copy(pdf,paste(folder.results,"cluster_grid.pdf",sep="/"), width=11,height=8)
dev.off()
```

```
# graph of k vs BIC values
df <- melt(mat)
colnames(df)[1:3] <- c("Group", "K", "BIC")
df$K <- as.factor(df$K)

(p1 <- ggplot(df, aes(x = K, y = BIC))+
geom_boxplot()+
theme_bw()+
xlab("Number of groups (K)")
)


# Let's see how diffferent the resulting clusters are
# and if we may have chosen too high value for K.

# choose an interval of n.da values
k <- 2:5

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

for(i in 1:length(dapc_l)){
  set.seed(9)
  grp_l[[i]] <- find.clusters(data, n.pca = 100, n.clust = k[i])
  dapc_l[[i]] <- dapc(data, pop = grp_l[[i]]$grp, n.pca = 100, n.da = k[i])
}

# A nice perspective is to create a scatterplot based on the
# discriminant functions. This helps us see how diffferent the
# resulting clusters are and if we may have chosen too high of
# a value for K.

# for K = max chosen
df <- as.data.frame(dapc_l[[ length(dapc_l) ]]$ind.coord)
df$Group <- dapc_l[[ length(dapc_l) ]]$grp
head(df)

#------------------------------------------------------------
# for the best k
K <- length(unique(grp$grp))-1
df <- as.data.frame(dapc_l[[ K ]]$ind.coord)
df$Group <- dapc_l[[ K ]]$grp
head(df)
pal <- RColorBrewer::brewer.pal(n=8, name = "Dark2")

#plot with no legend
(p3 <- ggplot(df, aes(x = LD1, y = LD2, color = Group, fill = Group))+
    geom_point(size = 4, shape = 21)+
    theme_bw()+
    labs(x = "PC1", y = "PC2")+
    scale_color_manual(values=c(pal))+
    scale_fill_manual(values=c(paste(pal, "66", sep = "")))+
    theme(legend.position = "none")
  )

#------------------------------------------------------------
# for k = 2

K <- 1
df <- as.data.frame(dapc_l[[ K ]]$ind.coord)
df$Group <- dapc_l[[ K ]]$grp
df %>% group_by(Group)
head(df)
pal <- RColorBrewer::brewer.pal(n=8, name = "Dark2")

(p2 <- ggdensity(df, x = "LD1",
          #add = "median",
          rug = FALSE,
          color = "Group",
          fill = "Group",
          palette = pal,
          label.rectangle = FALSE)+
```

```
    theme(legend.position = "none")
)



#-----------------------------------------------------------
# for k = 4
K <- 3
df <- as.data.frame(dapc_l[[ K ]]$ind.coord)
df$Group <- dapc_l[[ K ]]$grp
head(df)
pal <- RColorBrewer::brewer.pal(n=8, name = "Dark2")

#plot
(p4 <- ggplot(df, aes(x = LD1, y = LD2, color = Group, fill = Group))+
       geom_point(size = 4, shape = 21)+
       theme_bw()+
       labs(x = "PC1", y = "PC2")+
       scale_color_manual(values=c(pal))+
       scale_fill_manual(values=c(paste(pal, "66", sep = "")))+
    theme(legend.position = "none")
)

#-----------------------------------------------------------
# for k = 5
K <- 4
df <- as.data.frame(dapc_l[[ K ]]$ind.coord)
df$Group <- dapc_l[[ K ]]$grp
head(df)
pal <- RColorBrewer::brewer.pal(n=8, name = "Dark2")

#plot
(p5 <- ggplot(df, aes(x = LD1, y = LD2, color = Group, fill = Group))+
      geom_point(size = 4, shape = 21)+
      theme_bw()+
      labs(x = "PC1", y = "PC2")+
      scale_color_manual(values=c(pal))+
      scale_fill_manual(values=c(paste(pal, "66", sep = "")))+
    theme(legend.position = "none")
)

#-----------------------------------------------------------
# now look at barplots of the posterior probabilities of group
# assignment for each location

tmp <- as.data.frame(dapc_l[[1]]$posterior)
tmp$K <- k[1]
tmp$Individual <- rownames(tmp)
tmp <- melt(tmp, id = c("Individual", "K"))
names(tmp)[3:4] <- c("Group", "Posterior")
tmp$Region <- data$pop
df <- tmp

# 2:length, replace 2 by whichever K= you want to plot
# if you do not want to plot K=2, replace 2 by 3

for(i in 2:length(dapc_l)){
  tmp <- as.data.frame(dapc_l[[i]]$posterior)
  tmp$K <- k[i]
  tmp$Individual <- rownames(tmp)
  tmp <- melt(tmp, id = c("Individual", "K"))
  names(tmp)[3:4] <- c("Group", "Posterior")
  tmp$Region <- data$pop

  df <- rbind(df, tmp)
}

# build the ggplot
grp.labs <- paste("K =", k)
names(grp.labs) <- k

(p6 <- ggplot(df, aes(x = Individual, y = Posterior, fill = Group))+
    xlab("Individuals") + ylab("Posterior membership probability")+
```

```
       geom_bar(stat = "identity")+
        facet_grid(K ~ Region, scales = "free_x", space = "free",
                      labeller = labeller(K = grp.labs))+
        theme_bw()+
        #theme(legend.position='none')+ #group legend
        scale_fill_manual(values=c(pal))+
    theme(
          axis.text.x=element_blank(),
          axis.ticks.x=element_blank(),
          #axis.text.y=element_blank(),
          axis.ticks.y=element_blank())
  )

# we can now put all of this together into one plot

#tiff('dapc__k3_5_dapc.tiff', width=6.5, height=6.5, units='in', compression='lzw',
res=300)
ggarrange(ggarrange(p1,
                    p2,
                    p3,
                    p4,
                    p5,
                    ncol = 5, labels = c("A", "B","C","D","E")),
          p6,
          nrow = 2,
          labels = c("", "F"),
          heights = c(1, 2)
)


dev.copy(pdf,paste(folder.results,"Figure_2.pdf",sep="/"), width=11,height=8)
dev.off()
```