Ecological Genomics Tutorials: Population & Landscape Genomics 6

Setpember 27, 2023

Learning Objectives for 09/27/23

1. Review the diversity stats for our focal pops on the google doc
2. Estimate genetic differentiation (Fst) in ANGSD between our focal red spruce pops and black spruce
3. Visualize population structure using PCA and Admixture

1. Review the diversity stats

Let’s compare the diversity in our different pops on the google doc:

https://docs.google.com/spreadsheets/d/1y3GMvnGP65fYfBoJsdrixGLkGCe_TcDlo_7GFyPe1QM/edit?usp=sharing

I also put map in there for reference so you can see where different pops are located within the range.

- *What do you notice about the diversities?*
- *Where is Ne the highest/lowest?*
- *What do the average Tajima’s D values suggest about demographic history in these pops?*

2. Use ANGSD and the SFS for multiple pops to calculate genetic divergence between pops (Fst)

We can calculate Fst between any pair of populations by comparing their SFS to each other. For this, we’ll need to estimate the SFS for pairs of populations; we can each contribute to the overall analysis by looking at how our focal pop is divergent from the others.

- For this analysis, let’s calculate Fst between our focal red spruce population (MYPOP) and the black spruce samples…this could tell us which of our pops might be hybridizing. What would we expect for Fst in this case?

- Let’s write a bash script called ANGSD_Fst.sh that includes the following code:

```
# Start with the usual bash script header

```

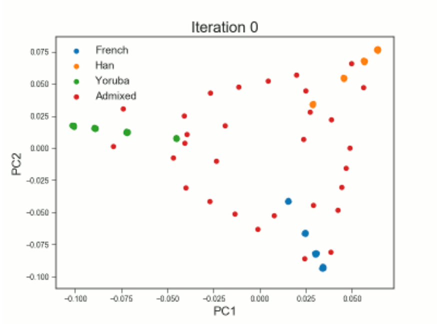
```
# Give yourself some notes

# Path to Black Spruce (BS) input saf.idx data:

BLKSPR="/netfiles/ecogen/PopulationGenomics/fastq/black_spruce/cleanreads/bam

OUTPUT=

MYPOP=""

cd ${OUTPUT}

# Estimate Fst between my red spruce pop and black spruce:

realSFS ${MYPOP}_.saf.idx \
        ${BLKSPR}/BS_all.saf.idx \
        -P 1 \
        >${MYPOP}_BS.sfs

realSFS fst index \
        ${MYPOP}_.saf.idx \
        ${BLKSPR}/BS_all.saf.idx \
        -sfs ${MYPOP}_BS.sfs \
        -fstout ${MYPOP}_BS \
        -whichFst 1

realSFS fst stats ${MYPOP}_BS.fst.idx
```

- Enter the *weighted* Fst value into the google sheet for your pop. What’s the trend?

3. Visualize popualtion structure across the landscape using PCA and Admixture

We often want to visualize differences in the genetic structure or genetic ancestry in our sample, and lots of papers we’ve read approach this using PCA or Admixture analysis. We can do each of these approaches on genotype likelihoods in ANGSD using a special routine called `pcANGSD`.

`pcANGSD` uses a really cool iterative approach where it refines the estimation of allele frequencies for each individual *at the same time* that it finds the clusters that individual may have ancestry within.



Here are some resources to understand the program options:

- The manual page
- A nice pcANGSD tutorial that walks through most of the routines
- The original paper describing the application to PCA and admixture are here: Meisner & Albrechtsen 2019, *Genetics*

Since we want to run `pcANGSD` for the entire set of samples – not just your focal pop – we need the genotype likelihoods from ANGSD for all 95 red spruce samples. That would take a long time to run (about 24 hrs) and would be redundant for each of you to do, so I ran these once for the class.

For your reference (and future work), the code I used to estimate the genotype likelihoods is here: (you don't have to run this now!)

`/netfiles/ecogen/PopulationGenomics/scripts/ANGSD_allRS_poly.sh`

and exported the genotype likelihoods in "beagle" format here:

`/netfiles/ecogen/PopulationGenomics/ANGSD/allRS_poly.beagle.gz`

We can use the beagle file containing the genotype likelihoods for all 95 red spruce samples as input to `pcANGSD`. The script is actually not too bad…let's give it a go:

```
# Start with the usual bash script header

# Give yourself some notes

# Path to your input data (where the beagle file lives)

INPUT=

# Path to save your output (in your home directory):

OUTPUT=

SUFFIX="allRS_poly"

# Make a copy of the list of bam files for all the red spruce samples and pla

cp ${INPUT}/allRS_bam.list ${OUTPUT}


# To run pcANGSD, you need to activate a "virtual environment" on the server

source /data/popgen/pcangsd/venv/bin/activate


# Then, run PCA and admixture scores with pcangsd:

pcangsd -b ${INPUT}/${SUFFIX}.beagle.gz \
        -o ${OUTPUT}/${SUFFIX} \
        -e 1 \
        --admix \
        --admix_alpha 50 \
        --threads 1
```

This will run pcANGSD assuming it fits a single "eigenvalue" to split your samples into K=2 clusters. If you want to explore higher levels of clustering in the future, you can include the `-e <int>` flag, where is a number that is K-1 number of clusters you want to fit.

Once the run is finished, use `FileZilla` to transfer the following files over to your repo on your laptop:

- allRS_bam.list
- allRS_poly.cov
- allRS_poly.admix.2.Q

When you have these files transferred (don't forget where you saved them to on your laptop!), open up RStudio and let's start making some figures!

Just a reminder, the following is R code, not bash. ;)

```
library(ggplot2) # plotting
library(ggpubr) # plotting

setwd("") # set the path to where you saved the pcANGSD results on your lapto

## First, let's work on the genetic PCA:

COV <- as.matrix(read.table("allRS_poly.cov")) # read in the genetic covarian
```

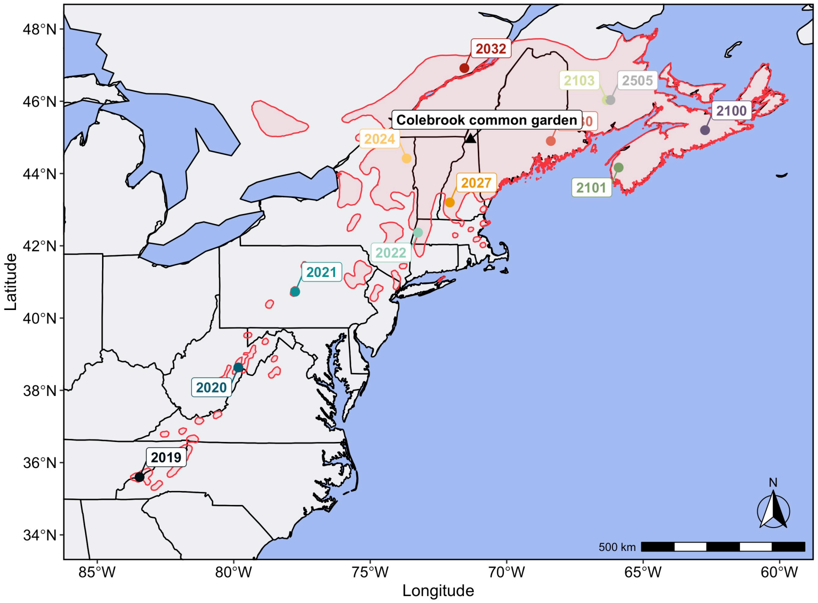
```
PCA <- eigen(COV) # extract the principal components from the COV matrix

## How much variance is explained by the first few PCs?

var <- round(PCA$values/sum(PCA$values),3)

var[1:3]

# A "screeplot" of the eigenvalues of the PCA:

barplot(var,
        xlab="Eigenvalues of the PCA",
        ylab="Proportion of variance explained")

## Bring in the bam.list file and extract the sample info:

names <- read.table("allRS_bam.list")
names <- unlist(strsplit(basename(as.character(names[,1])), split = ".sorted.
split = strsplit(names, "_")
pops <- data.frame(names[1:95], do.call(rbind, split[1:95]))
names(pops) = c("Ind", "Pop", "Row", "Col")

## A quick and humble PCA plot:

plot(PCA$vectors[,1:2],
     col=as.factor(pops[,2]),
     xlab="PC1",ylab="PC2",
     main="Genetic PCA")

## A more beautiful PCA plot using ggplot :)

data=as.data.frame(PCA$vectors)
data=data[,c(1:3)]
data= cbind(data, pops)

cols=c("#377eB8","#EE9B00","#0A9396","#94D2BD","#FFCB69","#005f73","#E26D5C",

ggscatter(data, x = "V1", y = "V2",
          color = "Pop",
          mean.point = TRUE,
          star.plot = TRUE) +
  theme_bw(base_size = 13, base_family = "Times") +
  theme(panel.background = element_blank(),
        legend.background = element_blank(),
        panel.grid = element_blank(),
        plot.background = element_blank(),
        legend.text=element_text(size=rel(.7)),
        axis.text = element_text(size=13),
        legend.position = "bottom") +
  labs(x = paste0("PC1: (",var[1]*100,"%)"), y = paste0("PC2: (",var[2]*100,"
```

```
  scale_color_manual(values=c(cols), name="Source population") +
  guides(colour = guide_legend(nrow = 2))




## Next, we can look at the admixture clustering:

# import the ancestry scores (these are the .Q files)

q <- read.table("allRS_poly.admix.2.Q", sep=" ", header=F)

K=dim(q)[2] #Find the level of K modeled

## order according to population code
ord<-order(pops[,2])

# make the plot:
barplot(t(q)[,ord],
        col=cols[1:K],
        space=0,border=NA,
        xlab="Populations",ylab="Admixture proportions",
        main=paste0("Red spruce K=",K))
text(tapply(1:nrow(pops),pops[ord,2],mean),-0.05,unique(pops[ord,2]),xpd=T)
abline(v=cumsum(sapply(unique(pops[ord,2]),function(x){sum(pops[ord,2]==x)}))
```

What have we learned about the genetic structure from the PCA and admixture plots?

- *What does the PCA seem to be telling us?*
- *What different picture does the admixture plot reveal? How does it relate to the PCA?*
- *Would we want to look for higher levels of K in the admixture analysis? How do we do that??*

For reference, here’s a map of the sample sites within the red spruce range:


2032
2103
2505
2100
Colebrook common garden
0
2024
2027
2101
2022
2021
2020
2019
N
500 km
48°N
46°N
44°N
42°N
40°N
38°N
36°N
34°N
Latitude
85°W
80°W
75°W
70°W
65°W
60°W
Longitude