Primer on Population Genetics

Michele Wyler

Institut für Virologie und Immunologie

July, 11th 2024

Abstract

Phylogenetic analyses are one of the most common bioinformatic procedures. They focus on identifying the relationships between individual genomic sequences. These analyses are commonly performed with both nucleotide and amino acid sequences. To explore these relationships on a genomic and/or individual level, researchers must adopt techniques from population genetics or its applied version known as breeding. This field of biology (or agronomy in the case of breeding) explores relationships of individuals, such as ancestry and mating patterns. Further important questions include the composition and structure of populations and the resulting dynamics. Common questions involve inferring population size and evaluating the impact of bottlenecks or bursts in individual numbers. Viruses can be explored using both phylogenetic and population genetics approaches. Unsegmented viruses, in particular, are simpler genetic sequences that can be analyzed using phylogenetics. On the other hand, they also function as individual members of a population that undergoes selection, adaptation, and speciation. The goal of this document is to introduce students without basic training and no analysis skills to essential techniques in population genetics, as well as basic programming and tools.

1. Distances

In this chapter we will explore the emergence of individual genotypes over time and geographic location. A phylogenetic tree can be viewed as a graphical depiction of a network of relationships defined through distances and topology. If topology is a complex matematical field, genetic distances can be easely represented as a distance matrix and analyzed using elementary linear algebry.

Biological samples can not only be described through the genetic distance buth on a variety of different ways. For example ecological niches can be described through species compostion or through geographic distances.

Mantel test allows to compare two different matrices with same dimension. In other words, it allows to statistically test the correlation of trends visible between two matrices. The Mantel test will deliver a p value. A low p value indicates that we can reject the hypothesis, that *matrix1* is unrelated to *matrix2*. We can conclude, that the two matrices are correlating.

```
# on R

# load libraries containing required functions
library(data.table)
library(vegan)

# 'Path' have to be adapted to display the path to each specific file
matrix1 <- fread('path')
matrix2 <- fread('path')
# Now you have loaded two different matrix: one containing the genetic distance and one with for exampl

# control if both objects have the same dimensions
dim(matrix1)
dim(matrix2)

# matrix need to be symmetric, if it's not the case you can produce them like this
#matrix2[lower.tri(matrix2)]  <- t(matrix2)[lower.tri(matrix2)]

# if the first column contains names, they need to be removed as follow
# rownames(matrix2) <- matrix2[, 1]
# matrix2[, 1] <- NULL

# the ordering of the matrices have to be the same
#orderingVector <- colnames(matrix1)
#matrix2 <- matrix2[orderingVector, orderingVector ]

# Once the files are loaded and formatted, we proceed with testing
mantelResults <- mantel(matrix1, matrix2,
                        method = "pearson",
                        permutations = 9999,
```

```
                    na.rm = TRUE)
mantelResults
```

2. Population structure

In this chapter, we will explore the possible structure of the observed population. We will utilize two common and well-known techniques, *Principal Component Analysis* (*PCA*) and *STRUCTURE* analysis. The outcomes of these two methods will allow for reliable identification of possible groupings within our population. We can therefore investigate if the two observed epidemiological waves are based on multiple genetic backgrounds.

We start by installing the famous PLINK suite, which will allow us to handle SNP formats efficiently. Furthermore, we will utilize the Admixture software, an implementation of the structure algorithm.

```
# on shell

# plink
wget https://s3.amazonaws.com/plink1-assets/plink_linux_x86_64_20231211.zip -O plink.zip
unzip plink.zip

# admixture
wget https://dalexander.github.io/admixture/binaries/admixture_linux-1.3.0.tar.gz -O admixture_linux-1.3
tar -xvzf admixture_linux-1.3.0.tar.gz
mv dist/admixture_linux-1.3.0/ admixture
```

2.1 Principal component analysis

Each individual can be defined through a set of single nucleotide polymorphisms (SNPs). With a PCA analysis, we try to reduce/collapse this set to a limited number of new variables, the so-called components. These new components are weighted summaries of the original SNP set, effectively capturing the most significant variation in the data. The PCA will try to separate the individuals as much as possible along each component.

```
# on shell

# plink doesn't allow for underscores
# if these is the case we can substitute these with a semicolon using sed as follow:
#sed -i 's/_/;/g' $TEMPDIR/new_jessica.vcf

plink --vcf file.vcf --pca --recode --make-bed --out plinkFiles
```

With this command we ask plink to convert the vcf file into various files (*–recode –make-bed*) and to calculate directly a PCA (*–pca*). By using *–out* we can specify the location and the first part of the name for all files. Plink will produce a variety of different files. In this chapter it's of interest the *.eigenvec file. The file will by default contain 20 columns representing the 20 first components.

```
# on R

library(data.table)
library(tidyverse)

# adapted the path to display the specific file
EIGENVAL <- 'path/to/plinkFiles.eigenvec'

tabella <- fread(EIGENVAL, data.table = F, sep = ' ')

# plot the resulting PCA
tabella %>%
  # define the axis of the plot (aka the first two components)
  ggplot(aes(x=V3, y=V4))+
  # specify that points are requested
  geom_point(size = 4)+
  # add explanatory labels
  labs(x='PC1', y='PC2')
```

2.2 STRUCTURE

In this chapter, we will infer the genetic structure of the analyzed population. By comparing each sample at the SNP level, we will test for the presence of distinct subpopulations. Furthermore, we aim to investigate the (not sampled) ancestry for each individual and the presence of admixture among them.

This approach is characterized by two distinct steps. First, we need to identify the optimal K value, representing the number of ancestral subpopulations. Through multiple tests of various K values, we can calculate the cross-validation errors for each number and select the lowest one. Further considerations for identifying the optimal K value could include *a priori* knowledge (in our case, epidemiological information) or parallel analyses (such as PCA performed in Chapter 2.1).

```
# on shell

# we use a shell scripting for loop
for NR in {1..10}; do
    admixture -j7 --cv $TEMPDIR/plink_MVV.bed $NR | tee admixture_K${NR}.log
done

# -j7 we run it using seven threats
# --cv ve are interested in calculating the cross validation error

# cross validation errors would usually be plotted, here we only look quickly at them
fgrep 'CV' admixture_K*.log
```

Once the best-fitting K is chosen, we can proceed with the second step of the analysis and estimate the ancestry of each sample. The *admixture* software will produce two files. Of our interest is the Q file, which will contain K columns describing each sample's composition.

```
# on shell

# example running on seven threats with K=2
admixture32 -j7 $TEMPDIR/plink.bed 2
```

Plotting has to be performed using an external script, as in this case with little custom R code.

```r
# On R

library(tidyverse)
library(data.table)



# select withc column of the plink tfam (1 or 2)
FAM <- "path/to/plinkFiles.tfam"
# Qfile of admixture
ADMIXTURE <- "path/to/plinkFiles.2.Q"

# First we need to get the labels of the samples
rowLabels <- fread(FAM, sep = ' ', header = F) %>%
  # select only the first column
  select(V1)
# change the name of the columns
colnames(rowLabels) <- c('Names')
# and convert them to factors (assign a value to a string, allows to keep the order)
rowLabels$Names <- as.factor(rowLabels$Names)



# Now we read the admixture table
matrice <- fread(ADMIXTURE, data.table = F)
# and add the sample informations from the previous step
matrice <- cbind(matrice, rowLabels)

# we perform a hierarhical clustering on the euclidean distance (to place similar samples togheter)
clustRows<- hclust(dist(matrice[, 1:ncol(matrice)-1]))
# and get the sample names (in order)
orderLabels <- matrice$Names[clustRows$order]



# plot
matrice %>%
```

```
  # we collapse the matrice to keep only one data point on each row
  gather(value = 'value', key = POP, -c(Names)) %>%
  # fix de order of the displayed samples
  mutate(Names = factor(Names, levels = orderLabels)) %>%
  ggplot(aes(fill = POP, y=value, x= Names)) +
  # instead of points we use histograms
  geom_bar(position="fill", stat="identity")+
  # define the order of the x axis...
  scale_x_discrete(labels = matrice$Names)+
  # ...and define few other graphical details
  theme_bw()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1, color = 'black'),
        legend.position = '')
```

3. Minimum spanning tree

Minimal spanning trees (MST) are a concept from graph theory. They aim to connect points with the minimum number of weighted edges. In the genomics field they are a valid alternative to classical phylogenecit trees in particular for organisms with clonal reproduction. We can consider viruses as part of these organism and therefore use MST to combine them in approximation of a infection chain.

Various implementation of the MST are available. Here we illustrate an example with the *emstreeR* package in R with successive plotting of the resulting MST.

```
# on R


# We use the same file as in chapter 2.1
EIGENVAL <- 'path/to/plinkFiles.eigenvec'
tabella <- fread(EIGENVAL, data.table = F, sep = ' ')
# alternatively a distance matrix can be used, then the dimension can be collapsed as follow
# tabella <- cmdscale(DISTmatrix)



# calulation of the MST
```

```
library(emstreeR)
out <- ComputeMST(tabella)
out


## plot with ggplot2:
library(ggplot2)
ggplot(out, aes(x = V1, y = V2, from = from, to = to))+
  geom_point()+
  # connect the points using the informations of the MST calculation
  stat_MST(colour="red")
```

4. Outlook

We dedicate this small tutorial to a brief introduction to basic population genetics techniques that can enhance our understanding of the diversity and adaptation of a viral population. The goal of this tutorial is to illustrate in a simple way the possibilities for data exploration and describe various complementary approaches.

The presented methods are used with default settings. A more in-depth analysis would require exploring the available options. Algorithms and functions are written to be broadly applicable; however, they may require specific fine-tuning to adapt to the data used.

A fundamental step in all diversity studies is the so-called "SNP filtering," the identification of the ideal SNP set on which to perform the analysis. This is why analyses based purely on qualitative nucleotide sequences are of dubious nature. Researchers must tackle this research question using a quantitative approach, allowing them to evaluate the consistency of each SNP. With PLINK, we introduce a common solution for SNP handling. Filtering for Minimum Allele Frequency (MAF), position, or quality is widely described in the literature.

The limited scope of this tutorial does not represent the extent of the entire research field. Population genetics is, in fact, a highly dynamic area, thanks to the advent of inexpensive genotyping technologies and increased computational power. A myriad of new approaches from computer science, mathematics, and molecular biology are currently being tested and evaluated. The applied aspect of this research field, known as breeding, aims to tackle global challenges such as climate change and increasing population, prioritizing

diversity studies. Here, we only indicate a few possible directions of current interest: coalescence theory, haplotyping, effects of recombination, mobile or ultra-conserved elements, and genomic reshufflings.

In conclusion, it is recommended to analyze a data set using a variety of different algorithms and approaches. A certain redundancy in the methods allows researchers to reliably assess the outcome of the experiments. Real trends will be visible across multiple methods, even if they vary in degree.