FST Exercise

Maria Izabel cavassim Alves

2018-02-12

## F-statistics from genotypic data:

In this exercise we will estimate the **Fst**, which is the expected level of heterozygosity in a population. The expected heterozygosity frequencies are drawn under the conditions of **Hardy-Weiberg Theorem**. It can be used to measure also the genetic variation across populations, by looking at the deviations of genetic diversity withing subpopulations () to the genetic diversity of the total population ():

9 equations will be used in this exercise:

1 Calculation of allele frequencies:

= $

2 Calculation of the expected genotypic counts under HWE:

* Expected AA =
* Expected Aa =
* Expected aa =

3 Calculate the local observed heterozygosity

4 Calculate the local expected heterozygosity

1. Calculate the local inbreeding coefficient of each subpopulation
2. Calculate (the frequency pf allele **A** over the total population)
3. Calculate (the frequency pf allele **a** over the total population)
4. Calculate the global heterozygosity indices ( over **I**ndividuals, **S**ubpopulations and **T**otal population)
5. based on observed heterozygosties in individuals in subpopulations:
6. based on **expected** heterozygosities in **subpopulations**:
7. based o expected heterozygosities for overall total population:
8. Calculate the Fst:

## Let's imagine that we have the following genotypes and genotype frequencies in each population:  
M <- as.matrix(rbind(c(125, 250, 125), c(50, 30, 20), c(100, 500, 400)))  
dimnames(M) <- list(c("Pop1", "Pop2", "Pop3"), c("AA","Aa", "aa"))  
  
print(M)

## AA Aa aa  
## Pop1 125 250 125  
## Pop2 50 30 20  
## Pop3 100 500 400

# What is the number of individuals genotyped in each population?  
N = rowSums(M)  
  
# Now let's calculate the allele frequencies in each population. Remember that the number of alleles is twice the number of genotypes.  
  
# Allele A:  
pop1\_p = (2\*M[1,1] + M[1,2])/(2\*N[[1]])  
  
# Allele a:  
pop1\_q = (2\*M[1,3] + M[1,2])/(2\*N[[1]])  
  
# Calculate the observed frequency of allele A for the other 2 populations  
  
  
# Now calculate the local observed heterozygosity:  
  
H\_obs\_pop1 = 250/500  
H\_obs\_pop2 = 30/100  
H\_obs\_pop3 = 500/1000  
  
# What would be the expected heterozygosity? Remember about the concepts of Hardy Weinberg equilibrium:  
H\_exp\_pop1 = 2\*pop1\_p\*pop1\_q  
  
# What is the local inbreeding coefficient of each population?  
#F\_s = (H\_exp - H\_obs )/ (H\_exp)  
  
F\_s\_pop1 = (H\_exp\_pop1 - H\_obs\_pop1) / H\_exp\_pop1  
  
# Now calculate the frequency of allele A over the total population (remember to mutiple the N frequencies):  
A\_s = (2\*colSums(M)[[1]] + colSums(M)[[2]])/(2\*sum(rowSums(M)))  
a\_s = (2\*colSums(M)[[3]] + colSums(M)[[2]])/(2\*sum(rowSums(M)))  
  
# Check if the calculations make sense. p + q must be equal to 1  
A\_s + a\_s

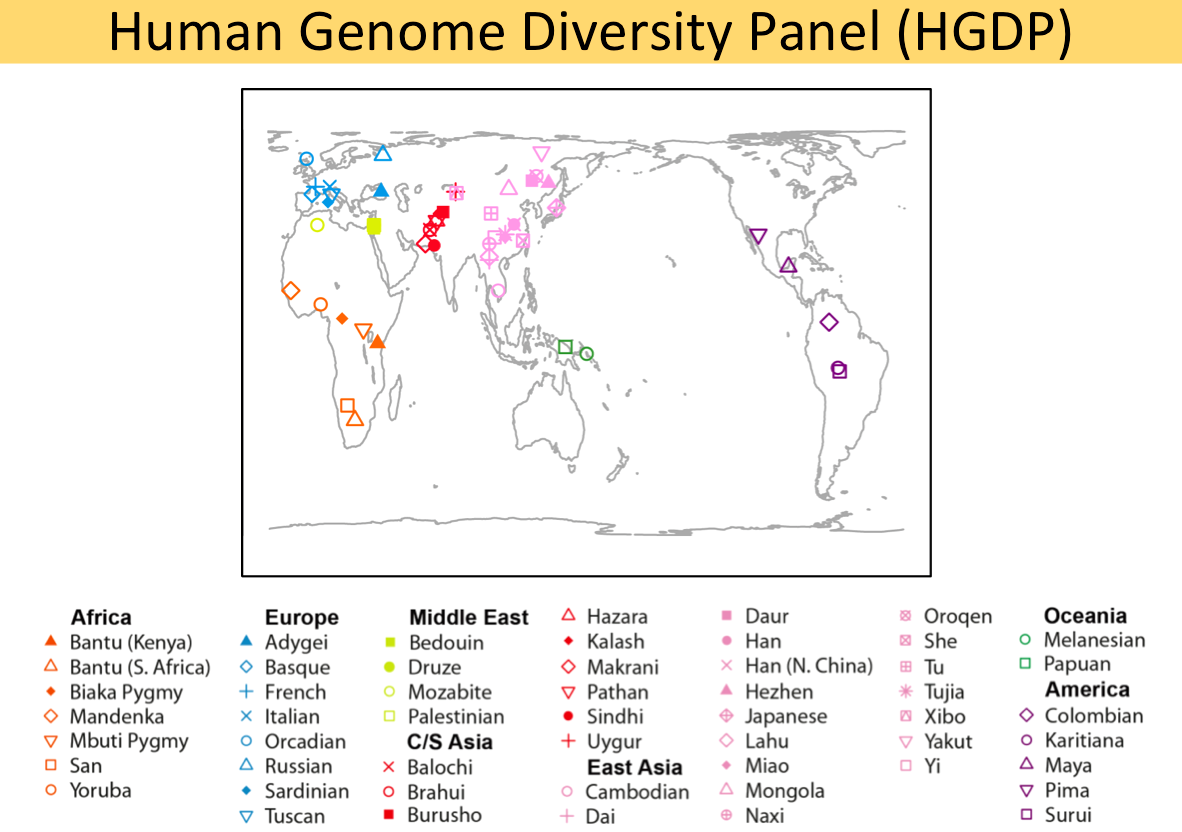
## [1] 1

# Now calculate the expected heterozygosity in subpopulations:

# Applying Fst in a real dataset:

## HGDP-CEPH Human Genome Diversity Cell Line Panel

In the next exercise we will be using the data from HGDP Panel, a consortium created to allow further research in human population genetics.  
A resource of 1063 lymphoblastoid cell lines (LCLs) from 1050 individuals in 52 world populations and corresponding milligram quantities of DNA is banked at the Foundation Jean Dausset-CEPH in Paris. These LCLs were collected from various laboratories by the Human Genome Diversity Project (HGDP) and CEPH in order to provide unlimited supplies of DNA and RNA for studies of sequence diversity and history of modern human populations. Information for each LCL is limited to sex of the individual, population and geographic origin.



Caption for the picture.

#With the HGDP dataset  
  
#install.packages('BEDASSLE')  
library(BEDASSLE)  
  
data(HGDP.bedassle.data)  
  
# Have a loog at what variables the data contain:  
?HGDP.bedassle.data()  
  
# How many populations does the data include?  
HGDP.bedassle.data$hgdp.metadata$Population

## [1] "Adygei" "Basque" "Italian" "French" "Orcadian"   
## [6] "Russian" "Sardinian" "Tuscan" "Bedouin" "Druze"   
## [11] "Mozabite" "Palestinian" "Balochi" "Brahui" "Burusho"   
## [16] "Hazara" "Kalash" "Makrani" "Pathan" "Sindhi"   
## [21] "Cambodian" "Dai" "Daur" "Han" "Hezhen"   
## [26] "Japanese" "Lahu" "Miao" "Mongola" "Naxi"   
## [31] "Oroqen" "She" "Tu" "Tujia" "Uygur"   
## [36] "Xibo" "Yakut" "Yi"

#Draw 2 populations at random from the Eurasian HGDP dataset  
 pop1 <- sample(nrow(HGDP.bedassle.data$allele.counts),1)  
 pop2 <- sample(nrow(HGDP.bedassle.data$allele.counts),1)  
  
#Calculate unbiased Fst between them  
 pairwise.Fst <- calculate.pairwise.Fst(  
 HGDP.bedassle.data$allele.counts[c(pop1,pop2),],  
 HGDP.bedassle.data$sample.sizes[c(pop1,pop2),]  
 )  
  
#Print that Fst to the console  
 print(sprintf("Fst between the %s population and the %s population is %s",  
 HGDP.bedassle.data$hgdp.metadata[pop1,1],  
 HGDP.bedassle.data$hgdp.metadata[pop2,1],  
 round(pairwise.Fst,3))  
 )

## [1] "Fst between the French population and the Hezhen population is 0.096"

# Now Calculate the Fst between populations from "French" and "Hezhen"  
 (pairwise.Fst <- calculate.pairwise.Fst(  
 HGDP.bedassle.data$allele.counts[c("French","Hezhen"),],  
 HGDP.bedassle.data$sample.sizes[c("French","Hezhen"),]  
 ))

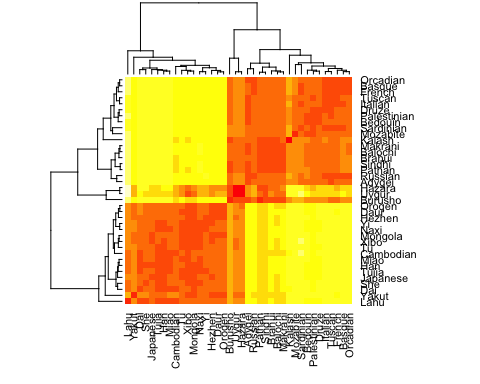
## [1] 0.09608346

#Calculate pairwise Fst between all population pairs  
 hgdp.pairwise.Fst <- calculate.all.pairwise.Fst(  
 HGDP.bedassle.data$allele.counts,  
 HGDP.bedassle.data$sample.sizes  
 )  
  
colnames(hgdp.pairwise.Fst) = HGDP.bedassle.data$hgdp.metadata$Population  
rownames(hgdp.pairwise.Fst) = HGDP.bedassle.data$hgdp.metadata$Population  
heatmap(hgdp.pairwise.Fst, names = HGDP.bedassle.data$hgdp.metadata$Population)

## Warning in plot.window(...): "names" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "names" is not a graphical parameter

## Warning in title(...): "names" is not a graphical parameter



#Plot pairwise Fst against geographic distance  
 plot(HGDP.bedassle.data$GeoDistance,  
 hgdp.pairwise.Fst,  
 pch=19,  
 col=HGDP.bedassle.data$EcoDistance+1,  
 ylab="pairwise Fst",  
 xlab="geographic distance",  
 main="isolation by distance")  
 legend(x="bottomright",pch=19,col=c(1,2),  
 legend=c("same side of Himalayas",  
 "opposite sides of Himalayas"))

