Methods for Detecting Intraspecific Natural Selection

Goal

Our goal is to explore approaches and methods, which seek to identify regions of the genome with signatures of natural selection. We will use real genomic data and two classes of tests: one based on population differentiation and another based on e extended haplotype homozygosity.

Dataset

Whole genome sequencing data by NGS (WG-NGS) from the 1000 Genomes Project phase III can be accessed through the link:
ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/

Data pre-processing

To optimize our time, we will analyze a pre-processed dataset for chromosome 2 corresponding to individuals sampled from the African (504 individuals), European (503 individuals), and East Asian (504 individuals) populations of the 1000 Genomes).

For now, repeating these filters is unnecessary, but here are the commands used.

- In vcftools software, remove the INDELs and singletons (~1h)

```
vcftools --gzvcf ALL.chr2.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz --remove-indels --min-alleles 2 --max-alleles 2 --maf 0.001 --max-maf 0.999 --recode --out SNPs_Chr2_filter
```

- In vcftools, select samples of individuals from the AFR, EAS and EUR populations (~ 30min) and filter to maf 0.05

```
vcftools --vcf SNPs_Chr2_filter.recode.vcf --keep pop_AFR_EAS_EUR_1000g.txt --min-alleles 2 --max-alleles 2 --maf 0.05 --max-maf 0.95 --recode --out SNPs_Chr2_AFR_EUR_EAS_maf
```

- In vcftools, select individual samples for each population

```
vcftools --vcf SNPs_Chr2_AFR_EUR_EAS_maf.recode.vcf --keep pop_AFR_1000g.txt --recode --out SNPs_Chr2_AFR_maf &

vcftools --vcf SNPs_Chr2_AFR_EUR_EAS_maf.recode.vcf --keep pop_EAS_1000g.txt --recode --out SNPs_Chr2_EAS_maf &

vcftools --vcf SNPs_Chr2_AFR_EUR_EAS_maf.recode.vcf --keep pop_EUR_1000g.txt --recode --outSNPs_Chr2_EUR_maf &
```

- Using vcftools, we estimate the Fst index between pairs of populations (~20 min each)

```
/vcftools --vcf ./dados/SNPs_Chr2_AFR_EUR_EAS_maf.recode.vcf --out AFR_EAS_maf --chr 2 --weir-fst-pop ./dados/pop_AFR_1000g.txt --weir-fst-pop ./dados/pop_EAS_1000g.txt &

vcftools --vcf ./dados/SNPs_Chr2_AFR_EUR_EAS_maf.recode.vcf --out AFR_EUR_maf --chr 2 --weir-fst-pop ./dados/pop_AFR_1000g.txt --weir-fst-pop ./dados/pop_EUR_1000g.txt &

vcftools --vcf ./dados/SNPs_Chr2_AFR_EUR_EAS_maf.recode.vcf --out EAS_EUR_maf --chr 2 --weir-fst-pop ./dados/pop_EAS_1000g.txt --weir-fst-pop ./dados/pop_EUR_1000g.txt &
```

Let's practice

Investigating a "Candidate Gene"

We refer to a "candidate gene" when we investigate whether there is evidence of natural selection in it based on previous results suggesting that it is a possible target for selection.

Detecting signatures of natural selection in the genome has the twofold meaning of (i) understanding which adaptive processes shaped genetic variation and (ii) identifying putative functional variants. In the case of humans, biological pathways enriched with selection signatures include pigmentation (Wilde et al. 2014), pathogen responses (Klunk et al. 2022; Couto-Silva, Nunes et al. 2023), and metabolic processes (Acuña-Alonzo et al. 2010).

The human Ectodysplasin A receptor gene, or *EDAR*, is part of the EDA signaling pathway which specifies prenatally the location, size, and shape of ectodermal appendages (such as hair follicles, teeth, and glands). *EDAR* is a textbook example of positive selection in East Asians (Sabeti et al. 2007) with genomic and functional experiments corroborating it. Also, genome-wide association studies found the same functional variant in EDAR associated hair morphology (Fujimoto et al. 2008) and incisor shape (Kimura et al. 2009) in East Asia populations and with several human facial traits (ear shape and chin protrusion) in Native American populations (Adhikari et al. 2016). Another plausible hypothesis stated that *EDAR* acted with *FADs* and *VDRs* genes in the Beringia Standstill (Hlusko et al. 2018), allowing these populations to survive in this extreme environment.

Natural Selection Tests

PART I

GENETIC DIFFERENTIATION AS EVIDENCE OF SELECTION (FST-BASED METHODS)

Through the exercises, discuss and answer the following questions:

1. The estimate of Fst by the Weir and Cockerham metric can sometimes generate negative values and "NA." What does that mean? How can this interfere with the results?
2. The Fst values observed between pairs of populations for the SNP rs3827760 (position 109513601) fall within which distribution quantiles of Fst values for the studied chromosome? Can they be considered outliers?
3. From the observed Fst values between population pairs and the significance estimates, what can we say about the rs3827760 SNP differentiation between populations?
4. Discuss how these results justify performing another type of analysis based on PBS (population branch statistics).
5. What does the PBS analysis reveal? What is the difference between PBS and FST analysis?

PRACTICE: FST

The files to download are at:
https://github.com/HunemeierLab/EMBO_Practical_Course_2024

1. Use R to run the following commands

a. Read the files with the Fst estimates (AFR_EUR.weir.fst, AFR_EAS.weir.fst and EAS_EUR.weir.fst)

```
names_header <- c("CHROM","POS","WEIR_AND_COCKERHAM_FST","NUM","DEN")

FST_AFR_EAS <- read.table("AFR_EAS.weir.fst", header=F, skip=1, col.names=names_header)

FST_AFR_EUR <- read.table("AFR_EUR.weir.fst", header=F, skip=1, col.names=names_header)

FST_EAS_EUR <- read.table("EAS_EUR.weir.fst", header=F, skip=1, col.names=names_header)
```

b. Eliminate duplicate positions

```
FST_AFR_EAS_filter <- FST_AFR_EAS[!duplicated(FST_AFR_EAS$POS),]

FST_AFR_EUR_filter <- FST_AFR_EUR[!duplicated(FST_AFR_EUR$POS),]

FST_EAS_EUR_filter <- FST_EAS_EUR[!duplicated(FST_EAS_EUR$POS),]
```

c. Take a look at the weir.fst file

```
head(FST_AFR_EAS_filter)

##   CHROM   POS WEIR_AND_COCKERHAM_FST      NUM     DEN
## 1     2 10554           0.318827 0.1388440 0.435482
## 2     2 10560           0.317773 0.1381040 0.434599
## 3     2 10566           0.315969 0.1373660 0.434744
## 4     2 10574           0.116785 0.0225713 0.193271
## 5     2 10587           0.368198 0.1540560 0.418407
## 6     2 10595           0.205058 0.0428390 0.208912

tail(FST_AFR_EAS_filter)

##       CHROM      POS WEIR_AND_COCKERHAM_FST       NUM     DEN
## 582959     2 243184391          0.00797051 0.00234889 0.294698
## 582960     2 243184570          0.04057770 0.01078470 0.265779
## 582961     2 243184927          0.23886500 0.05936500 0.248530
## 582962     2 243185274          0.23886500 0.05936500 0.248530
## 582963     2 243185679          0.01364920 0.00339660 0.248850
## 582964     2 243185846          0.21344200 0.04573750 0.214286
```

2. The estimation of FST by Weir and Cockerham can sometimes generate negative values, and Na. What does this mean? How can it interfere with the results?

a. Exclude positions whose FST result was equal to Na

```
FST_AfrEas_data <- FST_AFR_EAS_filter[-which(is.na(FST_AFR_EAS_filter[,3])),]

FST_AfrEur_data <- FST_AFR_EUR_filter[-which(is.na(FST_AFR_EUR_filter[,3])),]

FST_EasEur_data <- FST_EAS_EUR_filter[-which(is.na(FST_EAS_EUR_filter[,3])),]
```

b. The datasets now have different sets of SNPs, filter the files by overlapping the SNPs between them.

```
overlap_AfrEas_AfrEur <- FST_AfrEas_data[FST_AfrEas_data$POS %in% FST_AfrEur_data$POS,]

overlap_AfrEasEur_EasEur <- overlap_AfrEas_AfrEur[overlap_AfrEas_AfrEur$POS %in%
FST_EasEur_data$POS,]

FST_AfrEas_data_clean <- FST_AfrEas_data[FST_AfrEas_data$POS %in%
overlap_AfrEasEur_EasEur$POS,]

FST_AfrEur_data_clean <- FST_AfrEur_data[FST_AfrEur_data$POS %in%
overlap_AfrEasEur_EasEur$POS,]

FST_EasEur_data_clean <- FST_EasEur_data[FST_EasEur_data$POS %in%
overlap_AfrEasEur_EasEur$POS,]
```

c. Convert positions with estimates from FST < 0 to = 0

```
FST_AfrEas_data_clean[which(FST_AfrEas_data_clean[,3]<0),3] <- 0

FST_AfrEur_data_clean[which(FST_AfrEur_data_clean[,3]<0),3] <- 0

FST_EasEur_data_clean[which(FST_EasEur_data_clean[,3]<0),3] <- 0
```

3. Now with the data filtered and matched, let's check if the SNP rs3827760 (pos 109513601) is a candidate for natural selection.

a. Check if the SNP rs3827760, located at position 109513601, is an outlier in the FST distribution for any of the population pairs.

```
POS <- 109513601

FST_AfrEas_data_clean[FST_AfrEas_data_clean$POS==POS,]
```

```
##       CHROM      POS WEIR_AND_COCKERHAM_FST      NUM      DEN
## 266093     2 109513601              0.872881 0.762039 0.873016

FST_AfrEur_data_clean[FST_AfrEur_data_clean$POS==POS,]

##       CHROM      POS WEIR_AND_COCKERHAM_FST       NUM      DEN
## 266093     2 109513601           0.00997189 0.000108929 0.0109237

FST_EasEur_data_clean[FST_EasEur_data_clean$POS==POS,]

##       CHROM      POS WEIR_AND_COCKERHAM_FST      NUM      DEN
## 266093     2 109513601              0.859066 0.743056 0.864958
```

b. In which quartile of the distribution does the FST value for the SNP rs3827760 fall in each analyzed population pair?

```
FST_AfrEas_distr <- sort(FST_AfrEas_data_clean[,3])
FST_AfrEur_distr <- sort(FST_AfrEur_data_clean[,3])
FST_EasEur_distr <- sort(FST_EasEur_data_clean[,3])

FST_AfrEas_distrQT <- quantile(FST_AfrEas_distr, c(0.01, 0.05, 0.1, .25, .50,  .75, .90, 0.95, .99))
FST_AfrEas_distrQT

##          1%          5%         10%         25%         50%         75%
## 0.0000000000 0.0000079206 0.0031241100 0.0262112250 0.1056270000 0.2222135000
##         90%         95%         99%
## 0.3675160000 0.4685402500 0.6521344200

FST_AfrEur_distrQT <- quantile(FST_AfrEur_distr, c(0.01, 0.05, 0.1, .25, .50,  .75, .90, 0.95, .99))
FST_AfrEur_distrQT

##          1%          5%         10%         25%         50%         75%
## 0.000000000 0.000000000 0.002228373 0.018979525 0.081086300 0.186617500
##         90%         95%         99%
## 0.308151200 0.395242000 0.551998730

FST_EasEur_distrQT <- quantile(FST_EasEur_distr, c(0.01, 0.05, 0.1, .25, .50,  .75, .90, 0.95, .99))
FST_EasEur_distrQT

##          1%          5%         10%         25%         50%         75%
## 0.0000000000 0.0000000000 0.0004382527 0.0092101950 0.0476573500 0.1294852500
##         90%         95%         99%
## 0.2363264000 0.3175715000 0.4790830000
```

c. Please plot the FST values in a 10,000 base pair region adjacent to the SNP at position 109513601. Highlight the SNPs that are outliers in the 95th percentile in each population pair.

- Delimit the region of interest to 10000bp adjacent

```
SNPfrom_BP <- POS - 10000
SNPto_BP <- POS + 10000
```

- How many SNPs are in this bounded region?

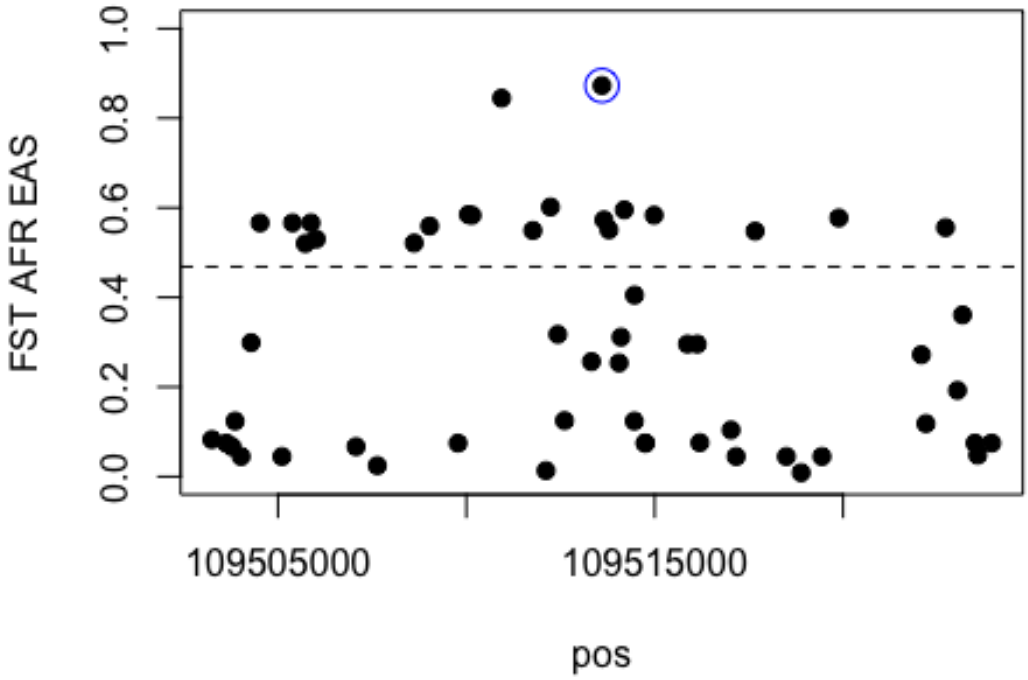
```
SNPfrom_id_AfrEas <- max(which(FST_AfrEas_data_clean[,2]<=SNPfrom_BP))
SNPto_id_Afr_Eas <- min(which(FST_AfrEas_data_clean[,2]>=SNPto_BP))
length(FST_AfrEas_data_clean[SNPfrom_BP:SNPto_BP, 2])

## [1] 20001

SNPfrom_id_AfrEur <- max(which(FST_AfrEur_data_clean[,2]<=SNPfrom_BP))
SNPto_id_Afr_Eur <- min(which(FST_AfrEur_data_clean[,2]>=SNPto_BP))
length(FST_AfrEur_data_clean[SNPfrom_BP:SNPto_BP, 2])

## [1] 20001

SNPfrom_id_EasEur <- max(which(FST_EasEur_data_clean[,2]<=SNPfrom_BP))
SNPto_id_EasEur <- min(which(FST_EasEur_data_clean[,2]>=SNPto_BP))
length(FST_EasEur_data_clean[SNPfrom_BP:SNPto_BP, 2])

## [1] 20001
```

- Select from the FST_AfrEur_data the region of interest

```
FSTdata_SNP_AfrEas <- FST_AfrEas_data_clean[SNPfrom_id_AfrEas:SNPto_id_Afr_Eas, ]
head(FSTdata_SNP_AfrEas)

##        CHROM      POS WEIR_AND_COCKERHAM_FST       NUM       DEN
## 266063     2 109503245             0.0826870 0.00859760 0.1039780
## 266064     2 109503631             0.0747407 0.00718525 0.0961357
## 266066     2 109503778             0.0668349 0.00590110 0.0882937
## 266067     2 109503862             0.1239490 0.02692580 0.2172320
## 266068     2 109504022             0.0448158 0.00616223 0.1375010
## 266069     2 109504287             0.2989630 0.12097500 0.4046480

FSTdata_SNP_AfrEur <- FST_AfrEur_data_clean[SNPfrom_id_AfrEur:SNPto_id_Afr_Eur, ]

FSTdata_SNP_EasEur <- FST_EasEur_data_clean[SNPfrom_id_EasEur:SNPto_id_EasEur, ]
```

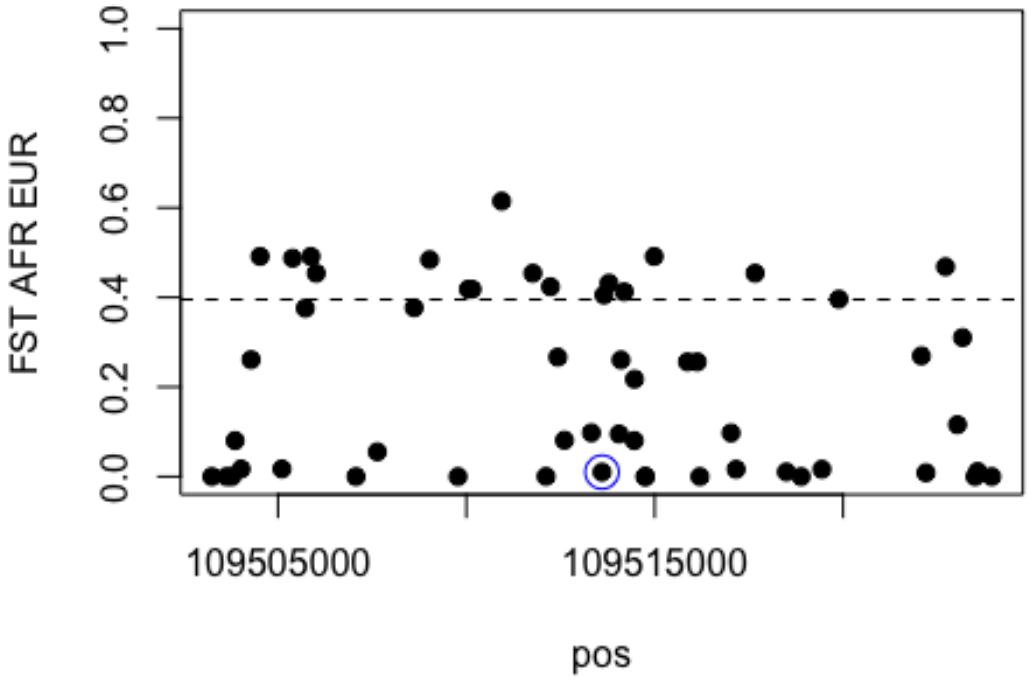
- PLOT
  - AfrEas

```
plot(ylim=c(0,1), x=FSTdata_SNP_AfrEas[,2], y=FSTdata_SNP_AfrEas[,3], xlab='pos', ylab='FST AFR EAS', pch=20, cex=1.5)

points(x=FSTdata_SNP_AfrEas[which(FSTdata_SNP_AfrEas[,2]==POS),2], y=FSTdata_SNP_AfrEas[which(FSTdata_SNP_AfrEas[,2]==POS),3], col='blue', cex=2)

abline(h=FST_AfrEas_distrQT[[8]], lty=2)
```



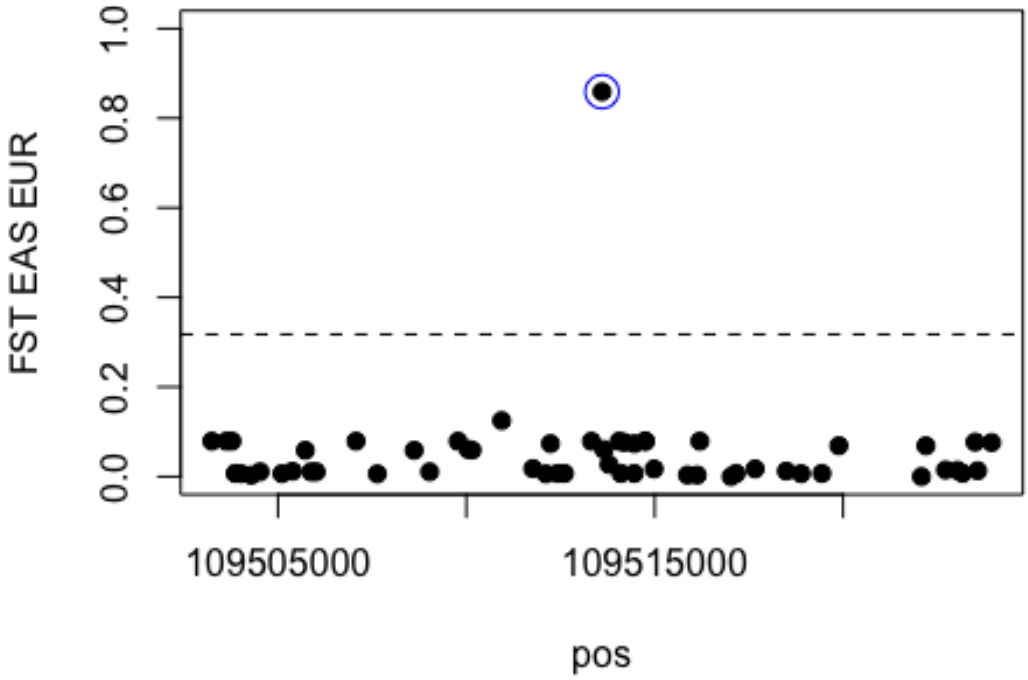
- AfrEur

```
plot(ylim=c(0,1), x=FSTdata_SNP_AfrEur[,2], y=FSTdata_SNP_AfrEur[,3], xlab='pos', ylab='FST AFR EUR', pch=20, cex=1.5)

points(x=FSTdata_SNP_AfrEur[which(FSTdata_SNP_AfrEur[,2]==POS),2], y=FSTdata_SNP_AfrEur[which(FSTdata_SNP_AfrEur[,2]==POS),3], col='blue', cex=2)

abline(h=FST_AfrEur_distrQT[[8]], lty=2)
```



- Eas Eur

```
plot(ylim=c(0,1), x=FSTdata_SNP_EasEur[,2], y=FSTdata_SNP_EasEur[,3], xlab='pos', ylab='FST EAS EUR', pch=20, cex=1.5)

points(x=FSTdata_SNP_EasEur[which(FSTdata_SNP_EasEur[,2]==POS),2], y=FSTdata_SNP_EasEur[which(FSTdata_SNP_EasEur[,2]==POS),3], col='blue', cex=2)

abline(h=FST_EasEur_distrQT[[8]], lty=2)
```



4. Can the candidate SNP be considered an outlier in all populations? What is the interpretation of this result?

a. Estimate the p-value for the candidate SNP from the distribution of FST values for each population pair.

- AfrEas

```
p_value_out_FST_AfrEas <-
sum(FST_AfrEas_data_clean$WEIR_AND_COCKERHAM_FST>=FST_AfrEas_data_clean[FST_AfrEas_data_clean$POS==109513601,3])/nrow(FST_AfrEas_data_clean)
p_value_out_FST_AfrEas

## [1] 0.0004772173
```

- AfrEur

```
p_value_out_FST_AfrEur <-
sum(FST_AfrEur_data_clean$WEIR_AND_COCKERHAM_FST>=FST_AfrEur_data_clean[FST_AfrEur_data_clean$POS==109513601,3])/nrow(FST_AfrEur_data_clean)
p_value_out_FST_AfrEur

## [1] 0.8143784
```

- EurEas

```
p_value_out_FST_EasEur <-
sum(FST_EasEur_data_clean$WEIR_AND_COCKERHAM_FST>=FST_EasEur_data_clean[FST_EasE
ur_data_clean$POS==109513601,3])/nrow(FST_EasEur_data_clean)
p_value_out_FST_EasEur
```

```
## [1] 5.302414e-06
```

PRACTICE: Population Branch Statistics (PBS)

Use R to:

1. **Based on population differentiation methods, let's explore selection signals using the PBS (Population Branch Statistic) approach:**

$$PBS = \frac{((-log(1 - FST\ AB)\ + (-log(1 - FST\ AC))\ -\ (-log(1 - FST\ BC))}{2}$$

   a. Perform PBS test, using EAS as candidate population for selection

```
PBS_EAS <- ((-log(1-FST_AfrEas_data_clean$WEIR_AND_COCKERHAM_FST))+(-log(1-
FST_EasEur_data_clean$WEIR_AND_COCKERHAM_FST))-(-log(1-
FST_AfrEur_data_clean$WEIR_AND_COCKERHAM_FST)))/2
```

   b. Convert negative PBS values to 0.

```
PBS_EAS[which(PBS_EAS<0)] <- 0
```

   c. Add to the data.table with FST values, a new column with PBS values.

```
fst_pbs<-as.data.frame(cbind(FST_EasEur_data_clean$POS,
FST_AfrEas_data_clean$WEIR_AND_COCKERHAM_FST,
FST_AfrEur_data_clean$WEIR_AND_COCKERHAM_FST,
FST_EasEur_data_clean$WEIR_AND_COCKERHAM_FST, PBS_EAS), stringsAsFactors=FALSE)

head(fst_pbs)

##     V1      V2      V3       V4     PBS_EAS
## 1 10554 0.318827 0.334988 0.00000e+00 0.0000000000
## 2 10560 0.317773 0.333938 0.00000e+00 0.0000000000
## 3 10566 0.315969 0.333938 2.24832e-06 0.0000000000
## 4 10574 0.116785 0.132890 4.42702e-04 0.0000000000
## 5 10587 0.368198 0.355929 0.00000e+00 0.0096164573
## 6 10595 0.205058 0.204892 0.00000e+00 0.0001043992
```

   d. Check the PBS value for the candidate SNP.

```
pbs_EDAR<-fst_pbs[fst_pbs$V1==POS,]

pbs_EDAR

##           V1      V2       V3      V4  PBS_EAS
## 259224 109513601 0.872881 0.00997189 0.859066 2.006037
```

e. In which quartile of the distribution does the PBS value for the SNP rs3827760 fall?

```
PBS_distrQT <- quantile(PBS_EAS, c(0.01, 0.05, 0.1, .25, .50,  .75, .90, 0.95, .99))
PBS_distrQT # estimar os quantils

##         1%         5%        10%        25%        50%        75%        90%
## 0.00000000 0.00000000 0.00000000 0.00000000 0.02506831 0.10666149 0.23044290
##        95%        99%
## 0.33611959 0.60783061
```

f. Plot the PBS values in a 10,000 base pair region adjacent to the SNP at position 109513601. Highlight the SNPs that are outliers in the 95th percentile.

- Select 10000bp adjacent to candidate SNP

```
SNP_FROM <- POS - 10000
SNP_TO <- POS + 10000

SNPfrom_PBS <- max(which(fst_pbs[,1]<=SNP_FROM))
SNPto_PBS <- min(which(fst_pbs[,1]>=SNP_TO))
length(fst_pbs[SNPfrom_PBS:SNPto_PBS, 1])

## [1] 55
```

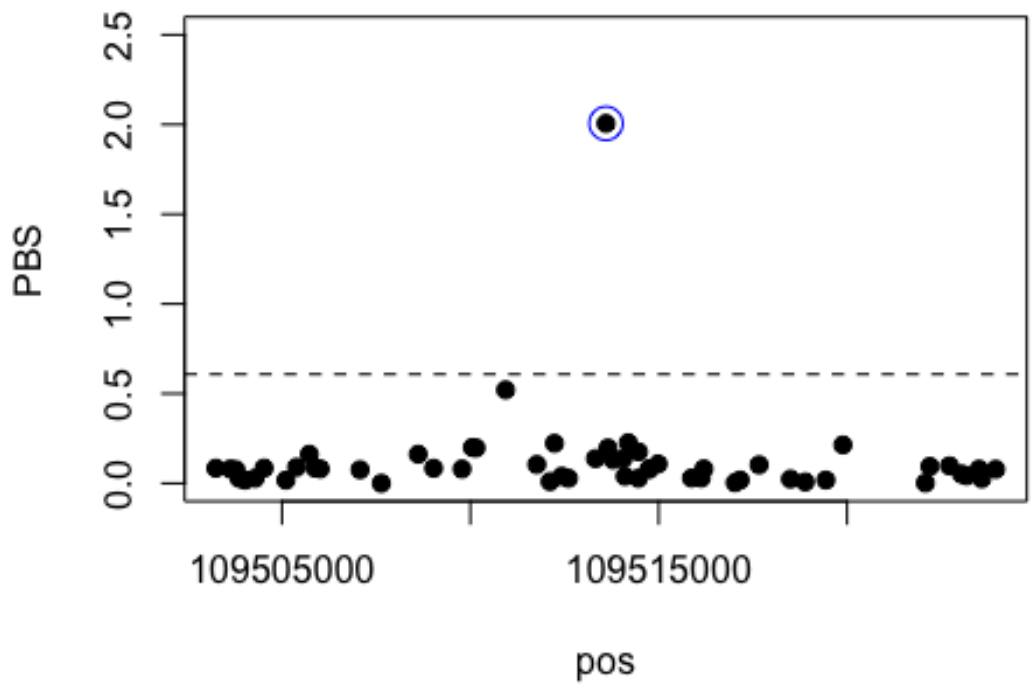
- Subset the candidate SNP region

```
subset_fst_PBS <- fst_pbs[SNPfrom_PBS:SNPto_PBS, ]
head(subset_fst_PBS)

##           V1       V2       V3        V4    PBS_EAS
## 259198 109503245 0.0826870 0.0000000 0.07902310 0.08431343
## 259199 109503631 0.0747407 0.0000000 0.07902310 0.08000079
## 259200 109503778 0.0668349 0.0000000 0.07902310 0.07574673
## 259201 109503862 0.1239490 0.0803297 0.00715918 0.02788793
## 259202 109504022 0.0448158 0.0171244 0.00664388 0.01762220
## 259203 109504287 0.2989630 0.2609700 0.00305409 0.02791831

subset_fst_PBS[subset_fst_PBS$V1==109513601,]

##           V1       V2       V3       V4  PBS_EAS
## 259224 109513601 0.872881 0.00997189 0.859066 2.006037
```

- Plot PBS values

```
plot(ylim=c(0,2.5), x=subset_fst_PBS[,1], y=subset_fst_PBS[,5], xlab='pos', ylab='PBS', pch=20,
cex=1.5)
```

```
points(x=subset_fst_PBS[which(subset_fst_PBS$V1==POS),1],
y=subset_fst_PBS[which(subset_fst_PBS$V1==POS),5], col='blue', cex=2)



abline(h=PBS_distrQT[[9]], lty=2) # plotar a linha
```



2. Based on FST and PBS tests applied so far, what can we conclude?

PART II

EXTENDED HAPLOTYPE HOMOZYGOSITY (EHH)

Different approaches are able to detect genomic signatures of selection at different timescales. More recent selection signals can be detected from the extended haplotype homozygosity approach.

With the following exercises, we seek to answer the following questions:

1) How is the haplotype profile of genetic variants under recent positive selection?

2) What is the profile of ancestral and derived haplotypes of the rs3827760 SNP in AFR and EAS?

3) The iHS score observed for the SNP rs3827760 fall within which distribution quantiles of iHS values for the studied chromosome? Can they be considered an outlier? How can we make this analysis more robust?

4) What information does the xp-EHH analysis add about natural selection in the candidate SNP?

PRACTICE: EHH

Use R to:

Install the rehh R package

```
install.packages("rehh")
```

Load rehh R package

```
library("rehh")
```

Use the following files

```
Chr2_EDAR_LWK_500K.recode.vcf #(African population)

Chr2_EDAR_CHS_500K.recode.vcf # (East Asian population)
```

The files to download are at:
https://github.com/HunemeierLab/EMBO_Practical_Course_2024

1. What is the profile of ancestral and derived haplotypes of the rs3827760 SNP in AFR and EAS?

a. Convert the data to haplohh format

```
data1<-data2haplohh(hap_file = "Chr2_EDAR_LWK_500K.recode.vcf", polarize_vcf = F,
vcf_reader = "data.table")

## * Reading input file(s) *
## Using package 'data.table' to read vcf.
## Extracting map information.
## Extracting haplotypes.
## Number of individuals which are
## Haploid Diploid Triploid, ... :
## 1 2
## 0 99
## * Filtering data *
## Discard markers genotyped on less than 100 % of haplotypes.
## No marker discarded.
## Data consists of 198 haplotypes and 29016 markers.
## Number of mono-, bi-, multi-allelic markers:
```

```
## 1 2
## 21289 7727

data2<-data2haplohh(hap_file = "Chr2_EDAR_CHS_500K.recode.vcf", polarize_vcf = F,
vcf_reader = "data.table")

## * Reading input file(s) *
## Using package 'data.table' to read vcf.
## Extracting map information.
## Extracting haplotypes.
## Number of individuals which are
## Haploid Diploid Triploid, ... :
## 1 2
## 0 105
## * Filtering data *
## Discard markers genotyped on less than 100 % of haplotypes.
## No marker discarded.
## Data consists of 210 haplotypes and 29016 markers.
## Number of mono-, bi-, multi-allelic markers:
## 1 2
## 24709 4307
```

b. Calculate the EHH for the candidate SNP (rs3827760) in AFR

```
ehh_calc_AFR<-calc_ehh(data1,mrk = "rs3827760")
ehh_calc_AFR

## An object of class "ehh"
## [[1]]
## [1] "rs3827760"
##
## [[2]]
## FREQ_A FREQ_D
##     1     0
##
## [[3]]
##            POSITION     EHH_A EHH_D
## rs552689611 109506332 0.05081270    0
## rs188449710 109506337 0.05081270    0
## rs191146014 109506387 0.05081270    0
## rs562458938 109506482 0.05081270    0
## rs182888230 109506509 0.05081270    0
## rs187445332 109506534 0.05081270    0
## rs113027039 109506633 0.05081270    0
## rs569087080 109506722 0.05081270    0
## rs561388021 109506760 0.05081270    0
## rs557899408 109506776 0.05081270    0
```

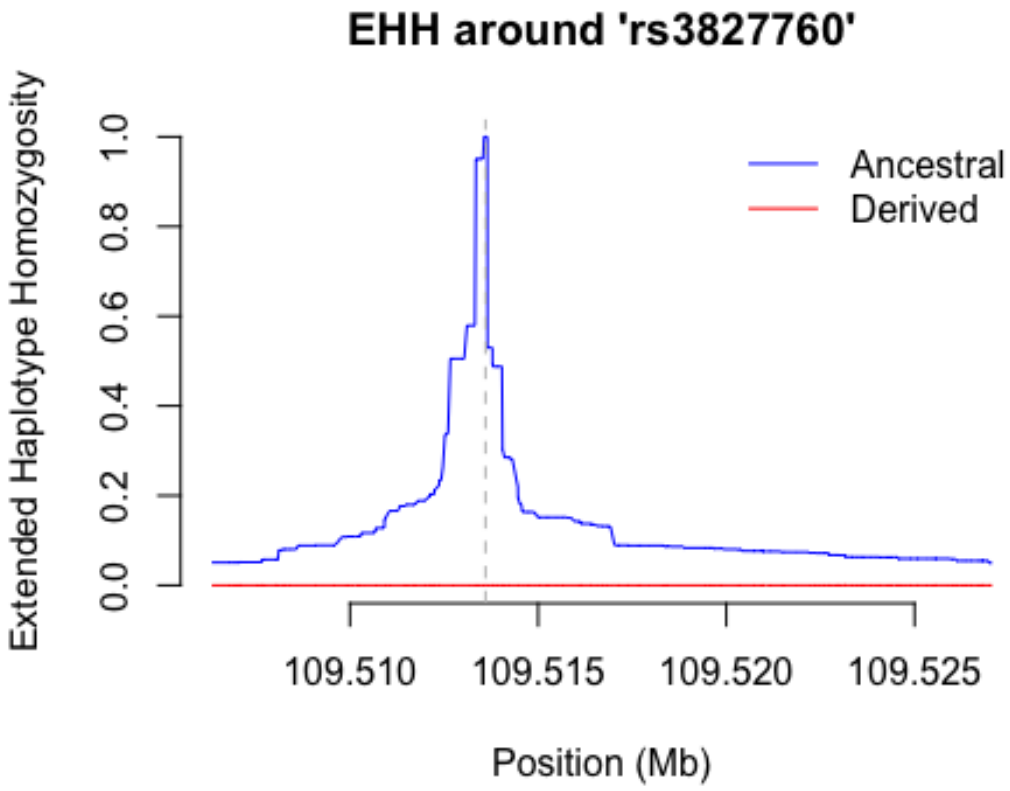
```
## rs535613872 109527004 0.05004358     0
## rs138769166 109527018 0.05004358     0
## rs201588688 109527031 0.05004358     0
## rs142670672 109527045 0.05004358     0
## rs535840595 109527050 0.05004358     0
##
## [[4]]
##    IHH_A    IHH_D
## 1767.692    0.000
```

c. Calculate the EHH for the candidate SNP (rs3827760) in EAS

```
ehh_calc_EAS<-calc_ehh(data2,mrk = "rs3827760")
ehh_calc_EAS

## An object of class "ehh"
## [[1]]
## [1] "rs3827760"
##
## [[2]]
##    FREQ_A    FREQ_D
## 0.0952381 0.9047619
##
## [[3]]
##             POSITION      EHH_A      EHH_D
## rs144341651 109411690 0.00000000 0.07234754
## rs546981503 109411706 0.00000000 0.07234754
## rs568352591 109411793 0.00000000 0.07234754
## rs529232887 109411800 0.00000000 0.07234754
## rs550870176 109412087 0.00000000 0.07234754
## rs569104695 109412193 0.00000000 0.07234754
## rs113683672 109412248 0.00000000 0.07234754
## rs79168135  109412332 0.00000000 0.07234754
## rs75277911  109412390 0.00000000 0.07234754
## rs147448762 109412402 0.00000000 0.07234754

## rs573491349 109586154 0.00000000 0.05519354
## rs534276564 109586174 0.00000000 0.05329992
## rs554546487 109586260 0.00000000 0.05329992
## rs574390523 109586275 0.00000000 0.05329992
## rs260694    109586313 0.00000000 0.05329992
## rs563073581 109586323 0.00000000 0.05329992
## rs534812588 109586359 0.00000000 0.05329992
##
## [[4]]
##     IHH_A     IHH_D
##  9979.001 55231.782
```

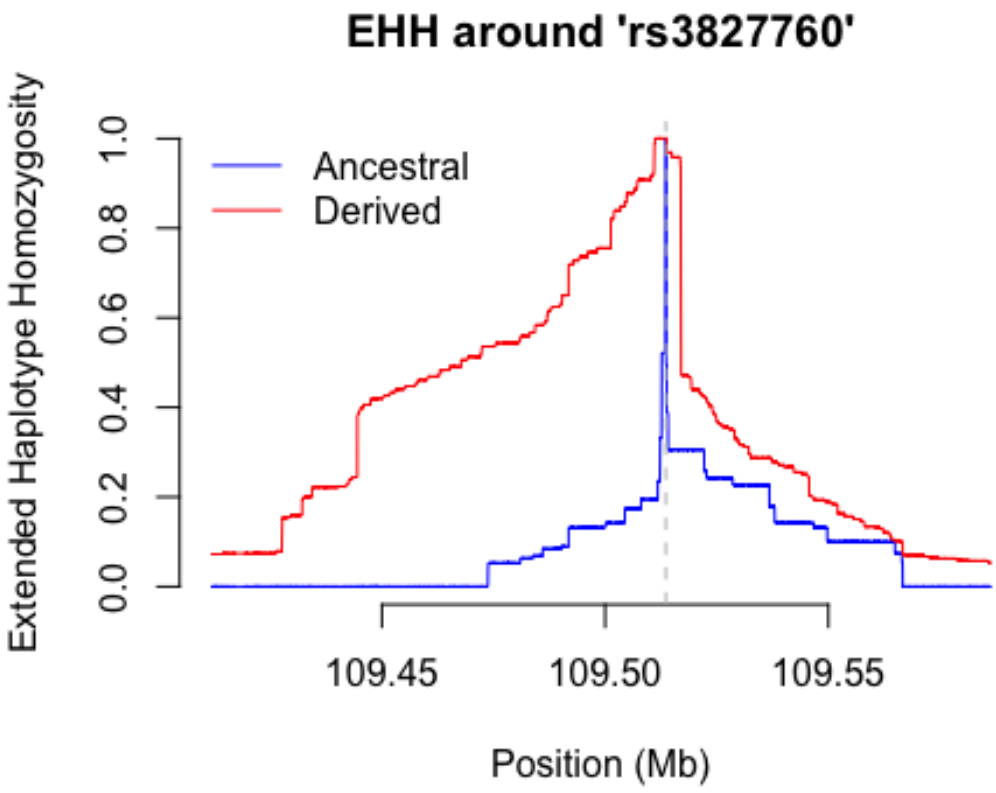
d. Plot EHH around “rs3827760” in AFR

```
plot(ehh_calc_AFR)
```



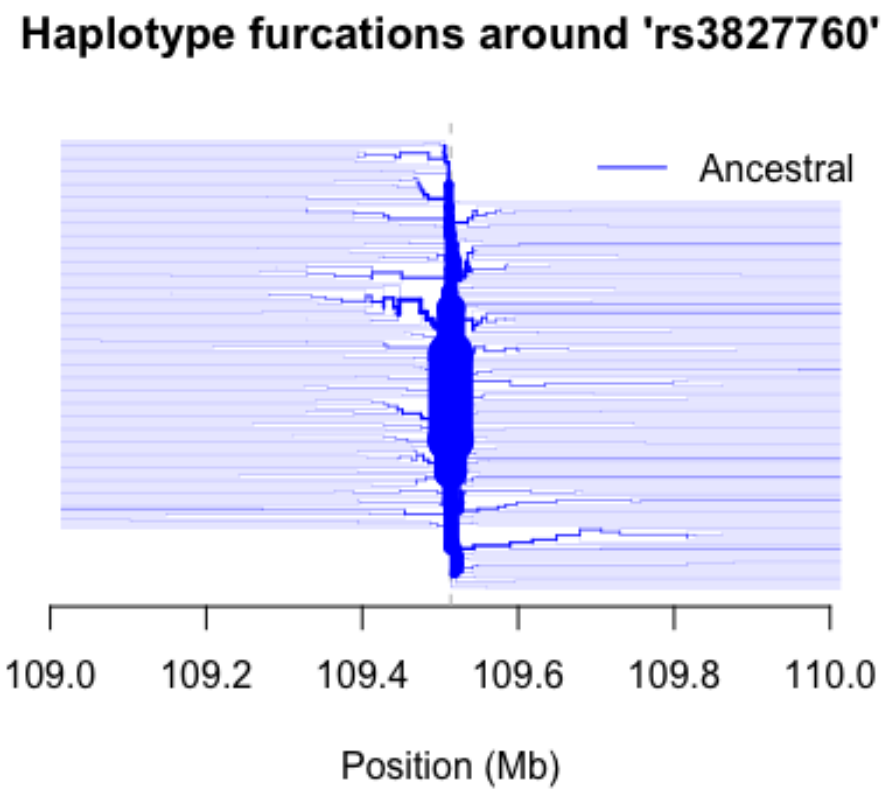
e. Plot EHH around “rs3827760” in EAS

```
plot(ehh_calc_EAS)
```



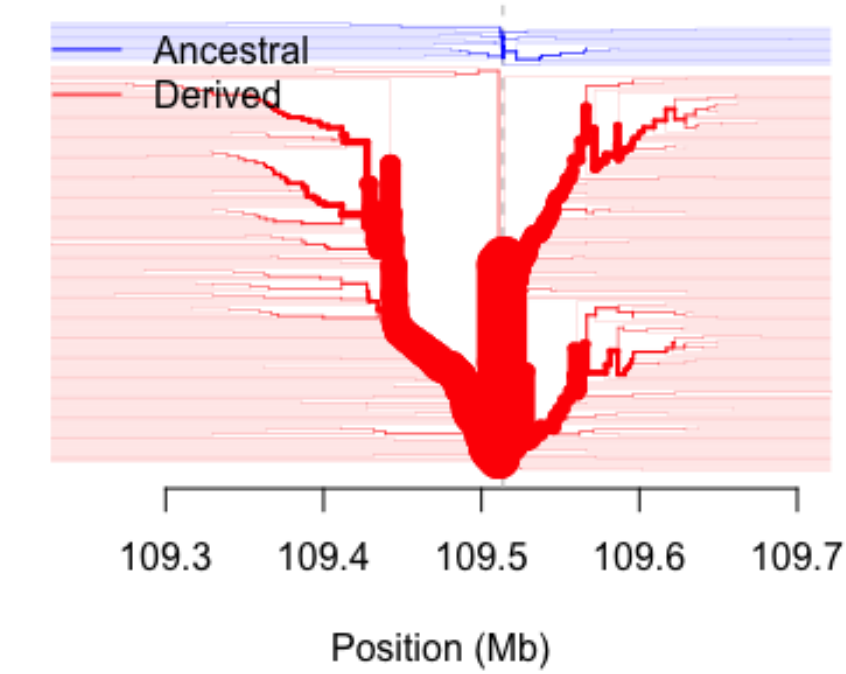
f. Calculate furcation trees around a candidate SNP in AFR

```
furcation<-calc_furcation(data1, mrk="rs3827760")
plot(furcation)
```



g. Calculate furcation trees around a candidate SNP in AFR

```
furcation<-calc_furcation(data2, mrk="rs3827760")
plot(furcation)
```


Haplotype furcations around 'rs3827760'
Ancestral
Derived
109.3
109.4
109.5
109.6
109.7
Position (Mb)

2. The integrated Haplotype Score (iHS) is a measure of the amount of extended haplotype homozygosity at a given SNP along the ancestral allele relative to the derived allele. This measure is typically standardized empirically to the distribution of observed iHS scores over a range of SNPs with similar derived allele frequencies.

a. Calculate the EHH for all SNPs in the file (~5min) for AFR

```
AFR<-scan_hh(data1)
```

b. Calculate the EHH for all SNPs in the file (~5min) for AFR

```
EAS<-scan_hh(data2)
```

c. Check eHH statistics for candidate SNP for AFR

```
AFR[AFR$POSITION==109513601,]

##         CHR  POSITION FREQ_A FREQ_D NHAPLO_A NHAPLO_D    IHH_A IHH_D      IES
## rs3827760   2 109513601     1     0      198       0 1767.692     0 1767.692
##            INES
## rs3827760 1767.692
```

d. Check eHH statistics for candidate SNP for AFR

```
EAS[EAS$POSITION==109513601,]

##         CHR  POSITION    FREQ_A    FREQ_D NHAPLO_A NHAPLO_D    IHH_A    IHH_D
## rs3827760   2 109513601 0.0952381 0.9047619       20      190 9979.001 55231.78
##            IES    INES
## rs3827760 43767.3 54737.7
```

e. Estimate the iHS in AFR (use min_maf = 0.02, freqbin = 0.01)

```
iHS.AFR<-ihh2ihs(AFR, min_maf = 0.02, freqbin = 0.01)
```

f. Estimate the iHS in EAS (use min_maf = 0.02, freqbin = 0.01)

```
iHS.EAS<-ihh2ihs(EAS, min_maf = 0.02, freqbin = 0.01)
```

g. Check the iHS score for the candidate SNP in AFR

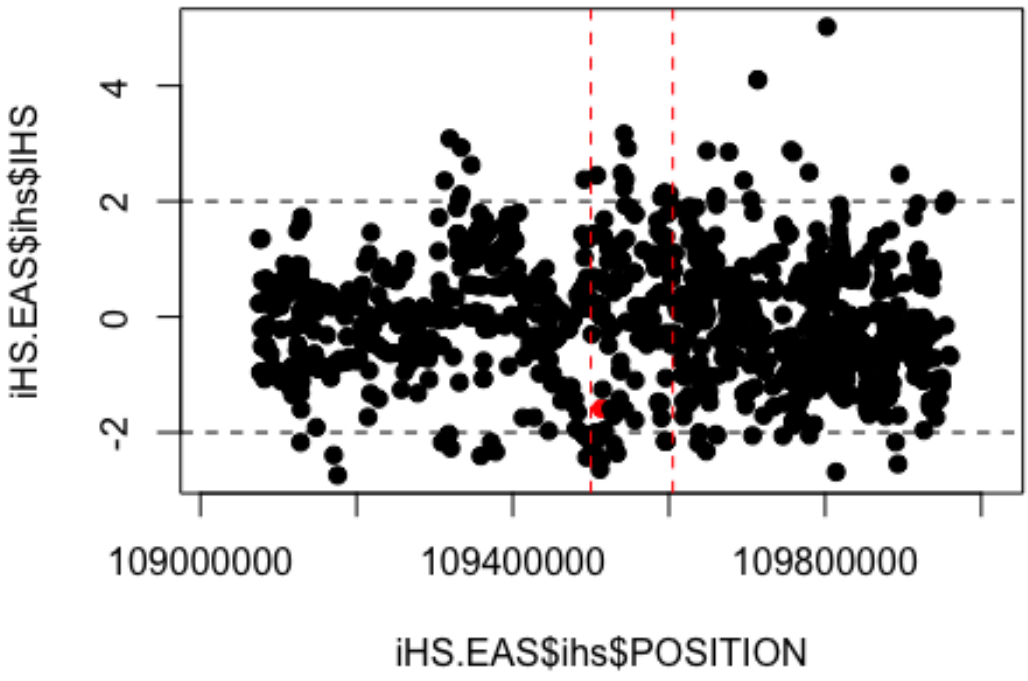
```
iHS.AFR$ihs[iHS.AFR$ihs$POSITION==109513601,]

## [1] CHR      POSITION  IHS      LOGPVALUE
## <0 rows> (or 0-length row.names)
```

h. Check the iHS score for the candidate SNP in EAS

```
iHS.EAS$ihs[iHS.EAS$ihs$POSITION==109513601,]

##         CHR  POSITION      IHS LOGPVALUE
## rs3827760   2 109513601 -1.588263 0.9499032
```

i. Plot the iHS score in EAS

```
plot(iHS.EAS$ihs$POSITION, iHS.EAS$ihs$IHS, col=ifelse(iHS.EAS$ihs$POSITION==109513601,
"red", "black"), pch=19)
abline(h=c(2,-2), lty=2 )
abline(v=c(109500000,109605000), col=c("red", "red"), lty=c(2,2), lwd=c(1, 1))
```



3. As we are looking at haplotypes, several individual SNPs have outlier values. One way to make the analysis more robust is to average a window of SNPs.

a. Create a function to estimate the mean in sliding windows.

```
slideFunct <- function(data, window, step){
 total <- length(data)
 spots <- seq(from = 1, to = (total - window + 1), by = step)
 result <- vector(length = length(spots))
 for(i in 1:length(spots)){
  result[i] <- mean(abs(data[spots[i]:(spots[i] + window - 1)]),na.rm=TRUE)
 }
 return(result)
}
```

b. Estimate the mean over a window of 50 SNPs with steps of 40 SNPs in EAS.

```
mean_iHS <- slideFunct(iHS.EAS$ihs$IHS, 50,40)
```

c. Identify the starting position of each window

```
slidePos <- function(data, window, step){
 total <- length(data)
 spots <- seq(from = 1, to = (total - window + 1), by = step)
 result <- vector(length = length(spots))
 for(i in 1:length(spots)){
  result[i] <- data[spots[i]]
 }
 return(result)
}

pos_wind_Eas <- slidePos(iHS.EAS$ihs$POSITION, 50,40)
```

d. Put the position information and average iHS in a table

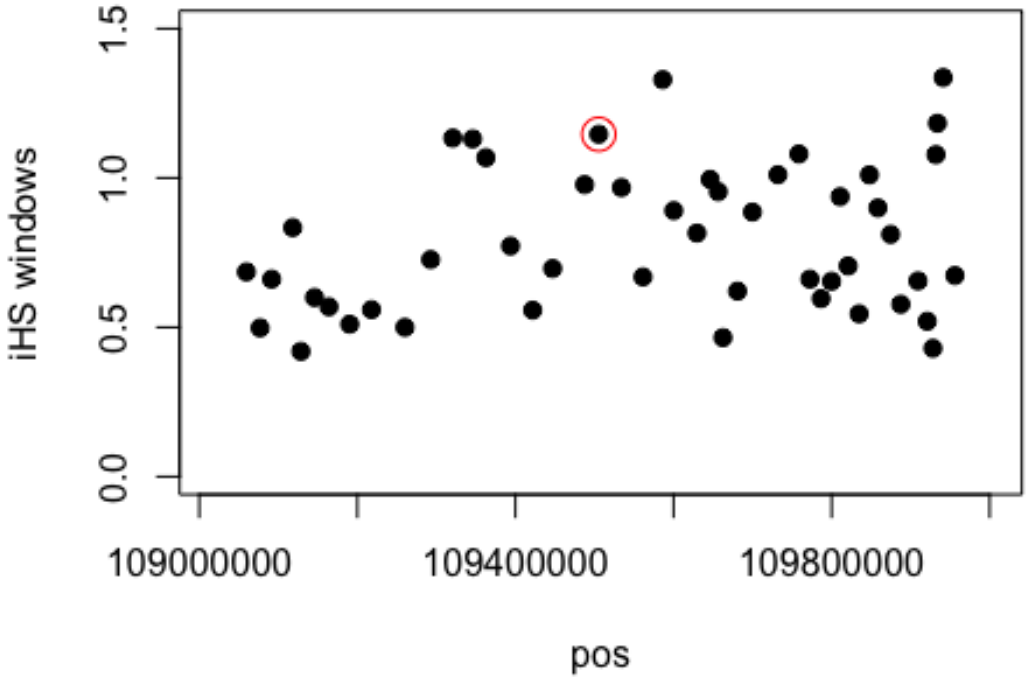
```
wind_iHS <- as.data.frame(cbind(pos_wind_Eas, mean_iHS), stringsAsFactors=FALSE)
```

e. Identify the window which contains the candidate SNP

```
Row_WIND_iHS <- wind_iHS[wind_iHS$pos_wind_Eas<=109513601,]
POS_WIND_iHS<-max(wind_iHS[nrow(Row_WIND_iHS),])
wind_iHS[wind_iHS$pos_wind_Eas==POS_WIND_iHS,]

##    pos_wind_Eas mean_iHS
## 21    109505388 1.146079
```

f. Plot the mean iHS per window

```
plot(ylim=c(0,1.5), x=wind_iHS[,1], y=wind_iHS[,2], xlab='pos', ylab='iHS windows', pch=20,
cex=1.5)
points(x=wind_iHS[which(wind_iHS[,1]==POS_WIND_iHS),1],
y=wind_iHS[which(wind_iHS[,1]==POS_WIND_iHS),2], col='red', cex=2)
```

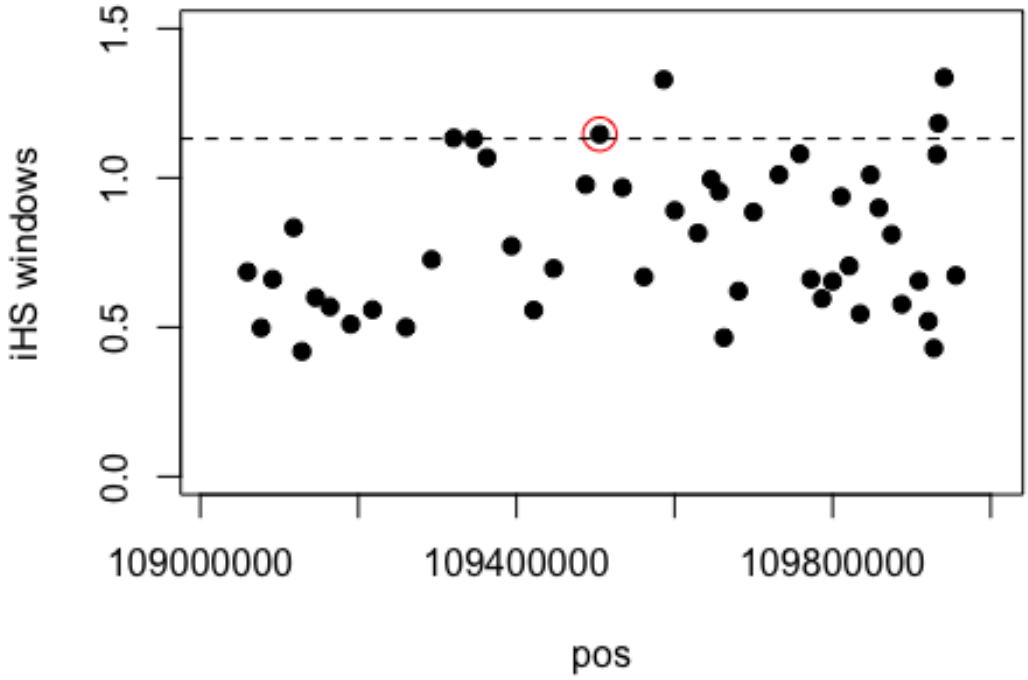


g. Check the distribution of iHS window in quantiles and check if the candidate SNP is an outlier.

```
windiHS_distrQT <- quantile(wind_iHS$mean_iHS, c(0.01, 0.05, 0.1, .25, .50,  .75, .90, 0.95, .99),
na.rm=T)
windiHS_distrQT

##       1%       5%      10%      25%      50%      75%      90%      95%
## 0.4240205 0.4766102 0.5071474 0.5913277 0.7159912 0.9820098 1.1314710 1.1701018
##      99%
## 1.3328222
```

h. Add the cut line for the quartile to the graph

```
plot(ylim=c(0,1.5), x=wind_iHS[,1], y=wind_iHS[,2], xlab='pos', ylab='iHS windows', pch=20,
cex=1.5)
points(x=wind_iHS[which(wind_iHS[,1]==POS_WIND_iHS),1],
y=wind_iHS[which(wind_iHS[,1]==POS_WIND_iHS),2], col='red', cex=2)
abline(h=windiHS_distrQT[[7]], lty=2)
```



4. What information does the xp-EHH analysis add about natural selection in the candidate SNP?

Cross-population extended haplotype homozygosity (xp-EHH) method was developed to detect selective sweeps in which the selected allele has approached or achieved fixation in one population but remains polymorphic in the other.

Our candidate SNP is not polymorphic in Africans, but for the purposes of the exercise, let's perform windowed xp-EHH analysis on SNPs adjacent to rs3827760.

a. Calculate the xp-EHH between EAS e AFR

```
xpEHH.EAS.AFR<-ies2xpehh(EAS,AFR)

## Scan of pop1 contains 29016 markers.
## Scan of pop2 contains 29016 markers.
## Merged data contains 29016 markers.
```

b. Calculate the average xp-EHH per 50 SNP window with 40 SNP steps

```
mean_xpEHH <- slideFunct(xpEHH.EAS.AFR$XPEHH, 50,40)
```

c. Identify the starting position of each window

```
pos_wind_Eas <- slidePos(xpEHH.EAS.AFR$POSITION, 50,40)
```

d. Put the position information and average xpEHH in a table

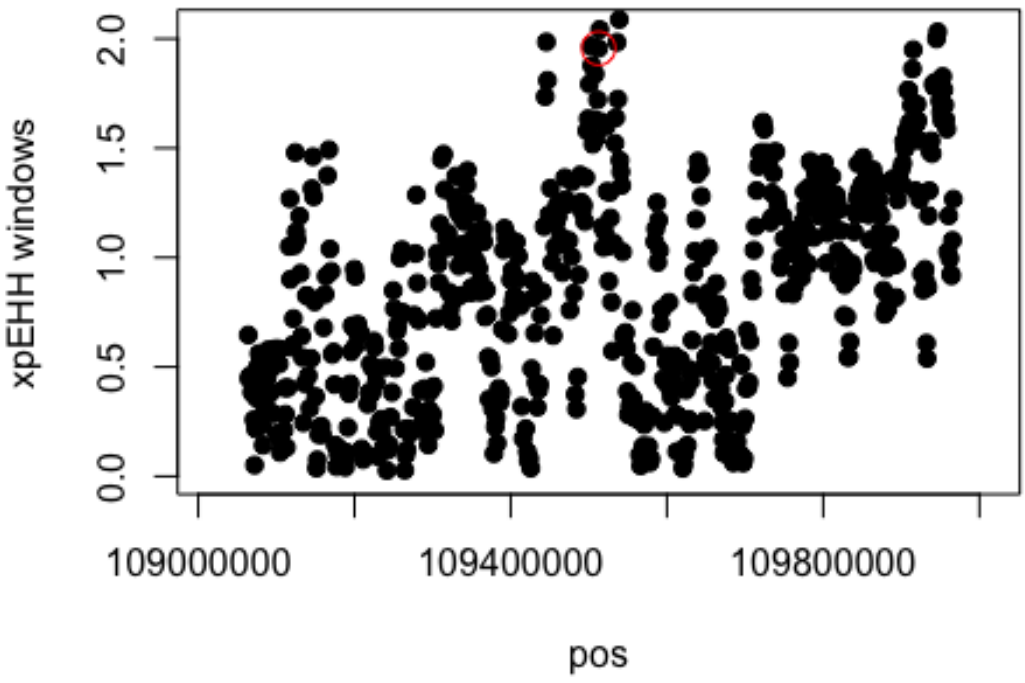
```
wind_xpEHH <- as.data.frame(cbind(pos_wind_Eas, mean_xpEHH), stringsAsFactors=FALSE)
```

e. Identify the window which contains the candidate SNP

```
Row_WIND_xpEHH <- wind_xpEHH[wind_xpEHH$pos_wind_Eas<=109513601,]
POS_WIND_xpEHH<-max(wind_xpEHH[nrow(Row_WIND_xpEHH),])
wind_xpEHH[wind_xpEHH$pos_wind_Eas==POS_WIND_xpEHH,]

##     pos_wind_Eas mean_xpEHH
## 344    109512468   1.956037
```

f. Plot the mean xpEHH per window

```
plot(ylim=c(0,2.05), x=wind_xpEHH[,1], y=wind_xpEHH[,2], xlab='pos', ylab='xpEHH windows',
pch=20, cex=1.5)
points(x=wind_xpEHH[which(wind_xpEHH[,1]==POS_WIND_xpEHH),1],
y=wind_xpEHH[which(wind_xpEHH[,1]==POS_WIND_xpEHH),2], col='red', cex=2)
```



g. Check the distribution of xpEHH window in quantiles and check if the candidate SNP is an outlier.

```
windxpEHH_distrQT <- quantile(wind_xpEHH$mean_xpEHH, c(0.01, 0.05, 0.1, .25, .50,  .75, .90,
0.95, .99), na.rm=T)
windxpEHH_distrQT

##       1%       5%      10%      25%      50%      75%      90%
## 0.04627655 0.10333280 0.16789186 0.42277950 0.88493217 1.23910534 1.47871451
##      95%      99%
## 1.67418581 1.99188441
```

h. Add the cut line for the quartile to the graph and outline the candidate gene region

```
plot(ylim=c(0,2.05), x=wind_xpEHH[,1], y=wind_xpEHH[,2], xlab='pos', ylab='xpEHH windows',
pch=20, cex=1.5)
points(x=wind_xpEHH[which(wind_xpEHH[,1]==POS_WIND_xpEHH),1],
y=wind_xpEHH[which(wind_xpEHH[,1]==POS_WIND_xpEHH),2], col='red', cex=2)

abline(h= windxpEHH_distrQT [[8]], lty=2)
abline(v=c(109500000,109605000), col="red")
```