Visualisation of genomic data and Admixture

We need additional R packages. Start R and give commands:

```
source("https://bioconductor.org/biocLite.R")
biocLite("biomaRt")
```

At this point you may need to close all other applications (or increase RAM and reboot). On my system the VM runs out of RAM if I don't close the browser.

When it asks to update other packages, answer "a" (all) and when it asks about installing locally, answer "y" (yes).

Do the same for other packages:

```
biocLite("Gviz")
biocLite("BiocGenerics")
```

Mean F_{st} as a measure of population distance

From Session 9:

------->

Genome-wide F_{st} is a measure of population divergence. We can compute that using vcftools.

```
cd ~/session_7
mkdir fst
```

Make a list of samples in two populations:

```
cat daf/popA.txt daf/popB.txt > fst/popAB.txt
```

Compute then the Weir's F_{st} for that pair:

```
bcftools view -S fst/popAB.txt data101_good.vcf.gz | vcftools --vcf -
--weir-fst-pop daf/popA.txt --weir-fst-pop daf/popB.txt --out fst/popAB

cat fst/popAB.log
```

Do this for each pair of populations. Which populations are closest to each other, which most distant? Do the results agree with the earlier results (PCA, ibs distance tree)?

<-------

Last time we looked at pairwise mean F_{st} between population as a measure of their divergence. If we do it for all populations, we need to correct for the mislabelling:

```
cd ~/session_7
cp daf/pop?.txt fst/

sed -i 's/popD-18/popI-18/' fst/popD.txt
sed -i 's/popI-18/popD-18/' fst/popI.txt
sed -i '/popB-34/d' fst/popB.txt
```

We can then compute all the pairwise Fst estimates and collect the results into a file:

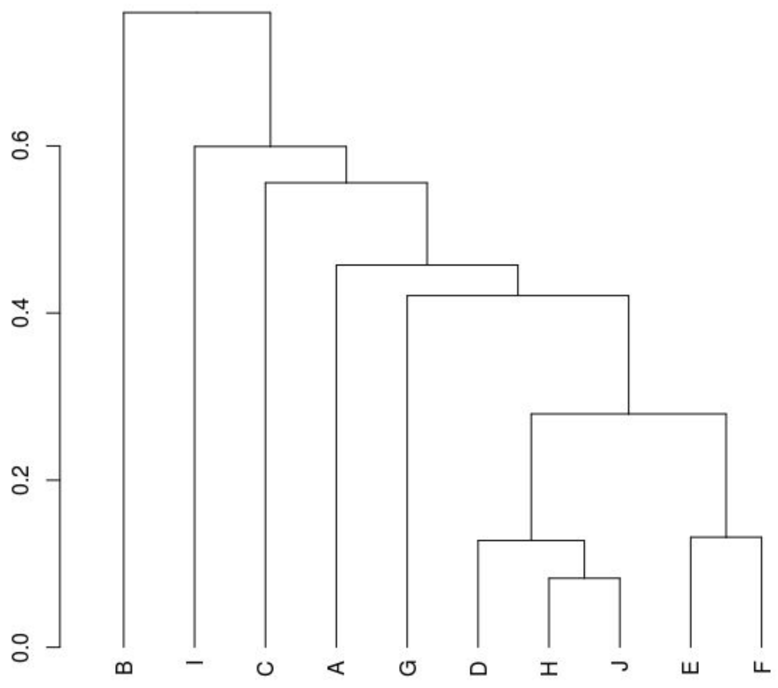
```
for i in A B C D E F G H I J; do
    for j in A B C D E F G H I J; do
        if [[ ! "$i" > "$j" ]] && [ "$i" != "$j" ]; then
            cat fst/pop$i.txt fst/pop$j.txt > fst/pop$i$j.txt;
            bcftools view -S fst/pop$i$j.txt data101_good.vcf.gz \
            | vcftools --vcf - --weir-fst-pop fst/pop$i.txt \
            --weir-fst-pop fst/pop$j.txt --out fst/pop$i$j;
        fi;
    done;
done


for i in A B C D E F G H I J; do
    for j in A B C D E F G H I J; do
        if [[ ! "$i" > "$j" ]] && [ "$i" != "$j" ]; then
            echo -n "$i $j "; grep -s mean fst/pop$i$j.log \
            | awk '{print $7}';
        fi;
    done;
done > fst/pairwise_fst.txt
```

We can get them recomputed from here:

```
 wget http://wasabiapp.org/vbox/data/session_10/pairwise_fst.txt -P fst
```

If we cluster the populations based on their F_{st} distances, we get a rather strange tree:

```
data=read.table("fst/pairwise_fst.txt",header=F)

mat=matrix(rep(0,100),10,10)
nms = c("A","B","C","D","E","F","G","H","I","J")
rownames(mat) = nms
colnames(mat) = nms

for(i in 1:45) {
   col = which(nms==data$V1[i])
   row = which(nms==data$V2[i])
   mat[col,row] = data$V3[i]
   mat[row,col] = data$V3[i]
}


d <- dist(mat, method = "euclidean")
fit <- hclust(d, method="average")
dend = as.dendrogram(fit)
plot(dend)
```



This tree doesn't agree with the cladogram that we obtained from the SNP data. The very high F_{st} values for population B appear to be a result of strong drift and the population's position in the tree above isn't correct. **This idea of using mean Fst values to create a population tree seems to be bad!**

Visualisation of genome-wide F_{st} estimates

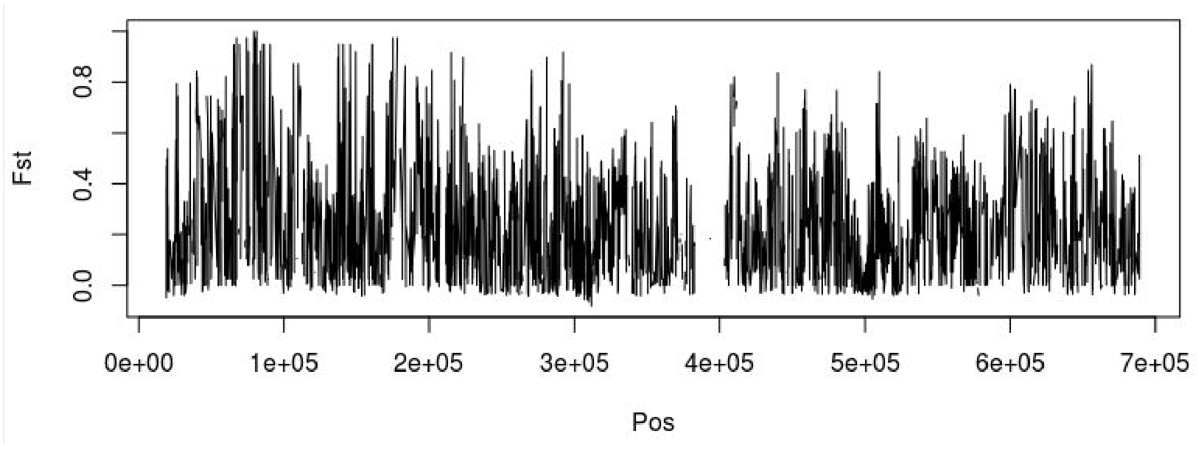
As discussed earlier, locally high F_{st} values can be an indication of differential selection. We don't have candidate regions but we can practise how to plot the data. First, let's compute F_{st} between two pond and two marine populations:

```
bcftools query -l ../session_3/data101_select1.vcf.gz|grep -e B -e C > fst/pond.txt
bcftools query -l ../session_3/data101_select1.vcf.gz|grep -e H -e J >
fst/marine.txt

cat fst/pond.txt fst/marine.txt > fst/pondmarine.txt
```

```
bcftools view -S fst/pondmarine.txt ../session_3/data101_select1.vcf.gz \
| vcftools --vcf - --weir-fst-pop fst/pond.txt --weir-fst-pop fst/marine.txt \
--out fst/pondmarine

less fst/pondmarine.weir.fst
```

We can now plot this in R:

```
d = read.table("fst/pondmarine.weir.fst",header=T)
ctg = d[d$CHROM=="ctg7180000005484",]

head(d)
head(ctg)
dim(d)
dim(ctg)

plot(ctg$POS,ctg$WEIR_AND_COCKERHAM_FST,xlab="Pos",ylab="Fst",type="l")
```



However, R can do **much** better than that.

We can use R package biomaRt to get genome data for three-spined stickleback and Gviz to make nice genome plots.

https://bioconductor.org/packages/release/bioc/html/biomaRt.html
https://bioconductor.org/packages/release/bioc/html/Gviz.html

These packages are not on the VM and need to be installed. This is explained above. An alternative is to do this on csc computers as explained below.

First we need to convert the Fst output into bed format. That can then be lifted-over to the three-spined genome coordinates:

```
awk '!/CHROM/{OFS="\t";print $1,$2,$2+1,$3,$1"."$2}' fst/pondmarine.weir.fst \
> fst/pondmarine_fst.bed

CrossMap.py bed reference/nineToThree.liftOver.gz fst/pondmarine_fst.bed \
fst/pondmarine_fst_threespine.bed
```

```
head fst/pondmarine_fst.bed
head fst/pondmarine_fst_threespine.bed
```

If problems, get the data from here:

```
wget http://wasabiapp.org/vbox/data/session_10/pondmarine_fst.bed
wget http://wasabiapp.org/vbox/data/session_10/pondmarine_fst_threespine.bed
```

Using R, we can see what are the names of Ensembl datasets for three-spined:

```
library(biomaRt)

ensembl=useMart("ensembl")
head(listDatasets(ensembl))
```

We see that the dataset is called "gaculeatus_gene_ensembl" and the version is "BROADS1". We also learn that the first contig is on positions 1:2582 in our data.

If you are using csc, do this first:

```
ssh username@taito.csc.fi
mkdir -p course/fst
cd course/fst
```

On your own computer:

```
scp fst/pondmarine_fst_threespine.bed username@taito.csc.fi:/homeappl/home/
username/course/fst
```

On csc:

```
module spider R
module load  gcc/4.7.2
module load  intelmpi/4.1.0
module load R/3.2.3
R


fst <- read.table("fst/pondmarine_fst_threespine.bed")
head(fst)
tail(grep("ctg7180000005484",fst$V5))
fst[2582,]
```

With that information we can build an R script:

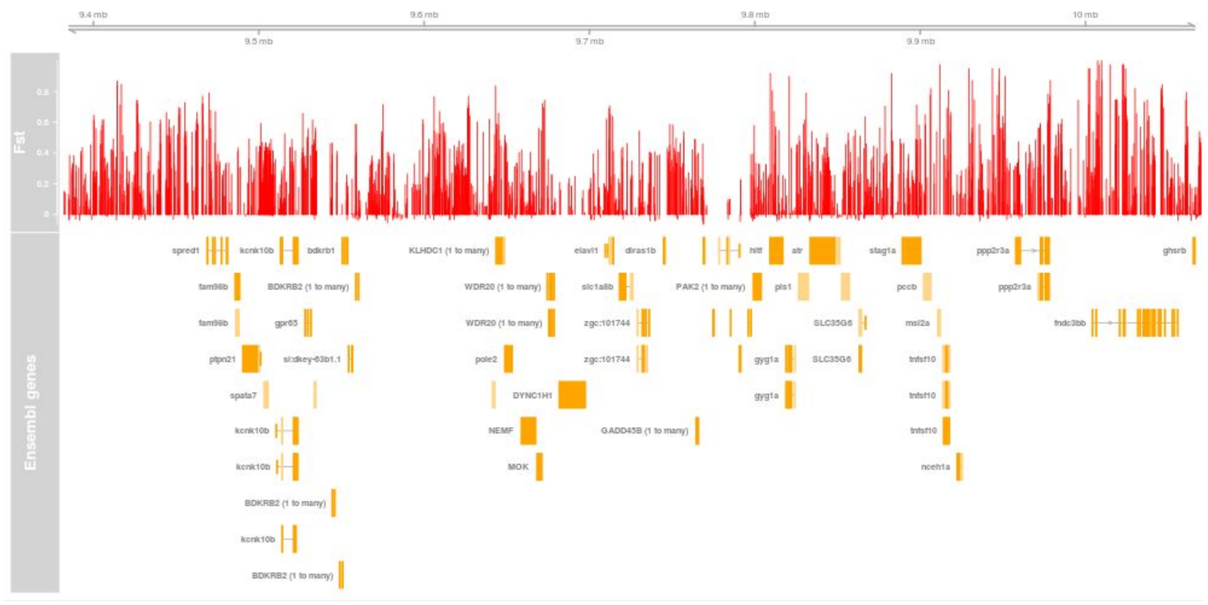
```
wget http://wasabiapp.org/vbox/data/session_10/plot_gviz.R -P fst
R -f fst/plot_gviz.R
```

on VM:

```
gpicview fst/gviz_plot.png
```

on csc:

```
eog fst/gviz_plot.png
```



The plot is nice but the information it contains may not be meaningful. For example, the lift-over does not preserve the SNP order and it needs more work to find the reasons for this (noise from repeats is not too serious; rearrangements can be problematic). In real analysis, one would clean the data more aggressively and use smoothing.

Admixture

Admixture (https://www.genetics.ucla.edu/software/admixture) is a faster version of program STRUCTURE and computes ML estimates of individual ancestries from multilocus SNP genotype datasets. Admixture input is .fam, .bim and .bed files. We created those for Plink but their format has to be changed a bit. (For example, Admixture requires that "chromosome names" are numbers.)

https://www.cog-genomics.org/plink2/formats#fam
https://www.cog-genomics.org/plink2/formats#bim
https://www.cog-genomics.org/plink2/formats#bed

```
cd ~/session_7
mkdir admixture
cd admixture


awk '{print $1" "$2" 0 0 "$5" -9"}' ../../session_3/plink/data101_select1.fam  \
> data101_select1.fam
```

```
perl -pe 's/ctg7180*//g;s/deg7180*//g' ../../session_3/plink/data101_select1.bim \
| awk '{$3=0; print $0}' > data101_select1.bim

cp ../../session_3/plink/data101_select1.bed .
```

See the difference between Plink and Admixture input.

```
head ../../session_3/plink/data101_select1.fam
head data101_select1.fam

head ../../session_3/plink/data101_select1.bim
head data101_select1.bim
```

For Admixture, we need to specify the number of ancestral populations (K) and the program then estimates how their contribute to each individual. Normally K is not known and the program is run with different values of K:

```
for K in `seq 2 10`; do admixture --cv data101_select1.bed $K | tee log${K}.out;
done &
```

With option "--cv" Admixture computes cross-validation error. The value of K with the lowest CV is considered the best explanation for the data:

```
grep CV log*
```

The program manual provides an example how to plot the result in R:

```
tbl=read.table("data101_select1.3.Q")
barplot(t(as.matrix(tbl)), col=rainbow(3), xlab="Individual #", ylab="Ancestry",
border=NA)
```

We can extend that to show the results for different values of K. Download the script:

```
wget http://wasabiapp.org/vbox/data/session_10/plot_admix.R
R -f plot_admix.R
```

This plot has the two samples swapped (can you guess which?) and one sample labelled incorrectly (which?). Uncomment the lines in the R script to correct the labels and plot again. Uncomment the line to use the real population names.