Exploratory statistics

PCA

We start with Principal Component Analysis using software "plink". As plink can do multiple analyses with one command, we also compute ibs distances and analyse them later.

First we convert the vcf data to plink's ped format using GATK. The conversion requires a metadata file. That could contain lots of information but in our case we don't even know the sexes of the samples.

https://www.broadinstitute.org/gatk/guide/tooldocs/org_broadinstitute_gatk_tools_walkers_variantutils_VariantsToBinaryPed.php

```
cd ~/session_3
mkdir plink

bcftools query -l data101_select1.vcf.gz | sort --version-sort \
| awk '{print $1" fid="$1";sex=0;phenotype=0"}' > plink/data101.meta

less plink/data101.meta

gatk -T VariantsToBinaryPed -V data101_select1.vcf.gz -m plink/data101.meta \
--bed plink/data101_select1.bed --bim plink/data101_select1.bim \
--fam plink/data101_select1.fam --minGenotypeQuality 0 -R reference/ninespine.fa
```

The second command fails as it requires TAB-formatted index. Do it first.

Now run plink:

```
plink --bfile plink/data101_select1 --distance 1-ibs square gz --pca \
--out plink/data101_select1 --allow-extra-chr

less -S plink/data101_select1.eigenval
less -S plink/data101_select1.eigenvec
```

Run with R:

```
vec = read.table("plink/data101_select1.eigenvec",sep=" ",header=F)
val = read.table("plink/data101_select1.eigenval",sep=" ",header=F)
nms = array(vec[,1])

clr = rep("black",101)

pop = c("popA","popB","popC","popD","popE","popF","popG","popH","popI","popJ")

for(i in 1:10) {
  clr[grep(pop[i],nms)]=rainbow(10)[i]
```

```
}

plot(vec[,4],vec[,3],col=clr,pch="*")


png("plink/pca1.png",800,800)
par(mfrow=c(4,4),mar=c(2.1, 2.1, 2.1, 2.1))
for(x in 3:6) {
  for(y in 4:7) {
    if(y<=x) { plot(0,type='n',axes=FALSE,ann=FALSE) }
    else {
       plot(vec[,y],vec[,x],col=clr,pch="*",xlab=x-2,ylab=y-2,cex=2)
    }
    if(x==3){ mtext(y-2,3,1) }
    if(y==7){ mtext(x-2,4,1) }
    if(x==4 && y==4) {
      legend("center", legend=c(pop), pch="*", col=c(rainbow(10),"black"),
        title="Population", ncol=1, bty="n",cex=1.5)
    }
    if(x==5 && y==4) {
      legend("center", legend=paste(1:5,": ", round(val[1:5,]*100/sum(val),2),"%",
        sep=""), title="Variance explained",ncol=1,bty="n",cex=1.5)
    }
  }
}
dev.off()
```

On the VM, you can use program "gpicview" to view image files:

```
gpicview plink/pca1.png
```

Do the same for "data101_select2.vcf.gz" and "data101_select3.vcf.gz". (Easiest done if you copy the script to a text editor and edit it there.) Write the pca plots to files "pca2.png" and "pca3.png". Compare the results.

Clustering

On R, regular R packages are installed with command:

```
install.packages("<packagename>")
```

You then have to select "(HTTP mirrors)" from the bottom and then a mirror server from somewhere nearby. Installation of bioconductor packages will be discussed later.

We used plink to compute IBS (Identity By State) distances. We used options "1-ibs", "square" and "gz" to output similarity (= 1-distance) as a square matrix (i.e. not triangular) and to compress the output. See what it produced:

```
less -S plink/data101_select1.mdist.id
less -S plink/data101_select1.mdist.gz
```

```
zcat plink/data101_select1.mdist.gz| column -t | less -S
```

Then produce a clustering tree of these using R:

```
library(dendroextras)
library(dendextend)

nms = read.table("plink/data101_select1.mdist.id",header=F,sep="\t")[,1]
dst = read.table("plink/data101_select1.mdist.gz",sep="\t", header=F,
col.names=nms, fill=T)

# Hierarchical Clustering
d <- dist(dst, method = "euclidean")

fit <- hclust(d, method="average")
dend = as.dendrogram(fit)

lab = labels(dend)
labels(dend) = as.vector(nms[lab])
lab = labels(dend)

clr = rep("black",101)

pop = c("popA","popB","popC","popD","popE","popF","popG","popH","popI","popJ")

for(i in 1:10) {
  clr[grep(pop[i],lab)]=rainbow(10)[i]
}

labels_colors(dend) = clr

pdf("plink/ibs1.pdf",5,10)
par(mar=c(3, 1, 2, 6),xpd=NA,cex=0.5,oma=c(1,1,1,1))
plot(dend,horiz=TRUE,cex=0.8)
dev.off()
```

On the VM, you can use program "evince" to view pdf files:

```
evince plink/ibs1.png
```

Do the same for "data101_select2.vcf.gz" and "data101_select3.vcf.gz". Write the plots to files "ibs2.png" and "ibs3.png". Compare the results.

Try doing the same analyses for an uncleaned data set. Remember that binary SNPs can be extracted with the following command:

```
bcftools view -m2 -M2 -V mnps,indels -Oz -o data101_bin_snp.vcf.gz data101.vcf.gz
```