HGDP Population Structure Workshop

Data-set

To gain some experience running some basic population genetic analyses, we will look at Illumina 650Y array data from the CEPH-Human Genome Diversity Panel. This sample is a global-scale sampling of human diversity with 52 populations in total.

The data were generated at Stanford and are available from the following link:

http://hagsc.org/hgdp/files.html.

Genotypes were filtered with a GenCall score cutoff of 0.25 and individuals with call rate <98.5% were removed. Beyond this, the set of individuals is filtered down to a set of 938 unrelated individuals and then the data are provided in plink's binary format (as files `H938.bed`, `H938.fam`, `H938.bim`).

Note about logistics

You may have been given a single tarball with this workshop, or you may have downloaded it from github. In either case, look in the `data` subdirectory for files. If the `data` directory is empty (or if you find it is missing any files referred to below, eg, the H938… or H938_Euro… files), then navigate to this link http://bit.ly/1aluTln and download all the files as a `.zip` and put them in the data directory.

The commands below assume you are doing the exercises in a sister subdirectory to `data`. As such, binary commands are often denoted as being run by typing `../bin/plink` and the data files referenced are in `../data` for example. You may want to modify these if you are working in other directories. You can also updating your PATH variable to make entering commands easier (look on the web to learn this).

I recommend creating one subdirectory for each section below using the `mkdir` command. Finally the `results` subdirectory contains nearly all the results and intermediate output, so if you have trouble with any part - go there and inspect the files and output figures to help guide you.

Subset the data for European populations only [read-only]

[NOTE : This next step is read-only. I have gone ahead and run the command and put the output in the `data` directory to save time].

In the `data` directory, the `H938.clst.txt` file contains the specifications of each individual's population id. Using an awk command we make a list of all the individuals who are in the 6 Europan populations of the HGDP (Sardinian, Basque, Italian, Adygei, Orcadian, French). Using plink we can extract just those individuals into a new set of plink files.

```
awk '$3=="Sardinian"||$3=="Basque"||$3=="Italian"||$3=="Adygei"||$3=="Orcadian"||$3=="French"\
  {print $0}' H938.clst.txt > Euro.clst.txt
plink --bfile H938 --keep Euro.clst.txt --make-bed --out H938_Euro
```

Filter SNPs in linkage disequilirium [read-only]

[NOTE: Likewise, to save time this step is read-only!]

We also prepare a version of each data set in which we filter out sites that are in linkage disequilibrium using plink's pairwise genotypic LD filtering commands.

```
plink --bfile H938 --indep-pairwise 50 10 0.1 --noweb
plink --bfile H938 --extract plink.prune.in --make-bed --out H938.LDprune --noweb
plink --bfile H938_Euro --indep-pairwise 50 10 0.1 --noweb
plink --bfile H938_Euro --extract plink.prune.in --make-bed --out H938_Euro.LDprune --noweb
```

The LD pruning step takes a little time, so I've provided the output files in the data subdirectory.

Exploring Hardy-Weinberg predictions

In this section, you will assess how well the genotypes at each SNP fit Hardy-Weinberg proportions. Given the population structure in this dataset, we might have a chance to observe the -Wahlund effect- in which the observed propotion of heterozygotes is less than expected due to hidden population structure (thought of another way, each sub-population is in a sense inbred, lowering the heterozygosity).

Using plink to get basic gentoype counts

To begin, run the plink `--hardy` command. It formally tests for departures from Hardy-Weinberg proportions. To keep the analysis simple, use the `--chr` command to limit the analysis to SNPs on chromosome 2.

```
plink --bfile ~/shared_data/HGDP_Data/H938 --hardy --chr 2 --noweb --out H938
```

Next, you are going to read the output of this command into R and visually explore the predictions of Hardy-Weinberg proportions.

First - we need to deal with a pesky aspect of plink in that it tests HWE in cases/controls and across the whole sample ("ALL"). We want to look at just across the whole sample (as we don't have cases/controls). The `grep` command will pull out only the output lines with "ALL" in the line.

```
grep ALL H938.hwe > H938.hwe_reduced
```

Plotting in R

Now, we will use the following R function to make a plot of the frequency of each genotype relative to its allele frequency (note: actually the code plots only a sampling of 3000 SNPs to avoid an overloaded plot). (Credits to Graham Coop for this function - see his lab blog http://gcbias.org/ as a reference).

```
plot.geno.vs.HW<-function(file,title=""){


    #read in the HW file from plink
        plink.hwe<-read.table(file,as.is=TRUE)

        names(plink.hwe)<-c("chr","SNP.id","which.inds","a1","a2",
          "genotype","obs.het","exp.het","HWE.pval")

        counts<-sapply(plink.hwe$genotype,function(x){as.numeric(strsplit(x,"/")[[1]])})
        counts<-t(counts)
        tot.counts<-rowSums(counts)
        geno.freq<-counts/tot.counts
        allele.freq<-(geno.freq[,1]+.5*geno.freq[,2])

        these.minor<-sample(1:nrow(geno.freq),3000)
        these.major<-sample(1:nrow(geno.freq),3000)
        ss.allele<-c(allele.freq[these.minor],1-allele.freq[these.major])
        ss.geno<-rbind(geno.freq[these.minor,],geno.freq[these.major,c(3,2,1)])
    # If you have adjustcolor library installed the following code is nice...
        #plot(ss.allele,ss.geno[,1],xlim=c(0,1),ylim=c(0,1),col=adjustcolor("red",0.1), xlab="allele fr
        #points(ss.allele,ss.geno[,3],xlim=c(0,1),ylim=c(0,1),col=adjustcolor("blue",0.1))
        #points(ss.allele,ss.geno[,2],xlim=c(0,1),ylim=c(0,1),col=adjustcolor("green",0.1))

    plot(ss.allele,ss.geno[,1],xlim=c(0,1),ylim=c(0,1),col="red", xlab="allele frequency",ylab="genotyp
        points(ss.allele,ss.geno[,3],xlim=c(0,1),ylim=c(0,1),col="blue")
        points(ss.allele,ss.geno[,2],xlim=c(0,1),ylim=c(0,1),col="green")

        smooth=1/5
        lines(lowess(ss.geno[,1]~ss.allele,f = smooth),col="black")
        lines(lowess(ss.geno[,3]~ss.allele,f = smooth),col="black")
        lines(lowess(ss.geno[,2]~ss.allele,f = smooth),col="black")

        x=1:1000/1000
        lines(x,x^2,lty=2)
        lines(x,2*x*(1-x),lty=2)
        lines(x,(1-x)^2,lty=2)
        legend(x=0.3,y=1,col=c("red","blue","green",rep("black",2)),
          legend=c("Homozygote AA","Homozygote aa","Heterozygote Aa","Mean","Hardy Weinberg Expectation
          pch=c(rep(1,3),rep(NA,2)),lty=c(rep(NA,3),1,2), cex=0.5, pt.cex = 1)
    }
```

You can use this function directly in R to make a plot or you can produce, for example, a png graphic file with the plot, as shown below.

```
png(file="HGDP_hwe.png")
plot.geno.vs.HW(file="H938.hwe_reduced",title="HGDP")
dev.off()
```

Questions

1. Do the genotypic frequencies roughly follow the basic patterns expected for Hardy-Weinberg proportions (e.g. P_{AA} is approximately quadratic in p)?
2. Looking more carefully, is the HW prediction for the proportion of heterozygotes given allele frequency generally too high or too low relative to the empirically observed values? What might explain the deviation?
3. Now, go through the same steps for the full H938_Euro set of plink files. Compare the deficiency in heterozygotes between the world-wide data and the European only data. In which is the deficiency smaller? Why might that be the case?

Allele frequency spectra

The allele frequency spectra is a count of the number of variant positions that have a particular allele frequency count (i.e. the "freqeuncy of different freqecies"!). This can be done using the `hist` function in R to make a histogram. The only trick is that there is a variable amount of missing data in the sample. As a way to avoid this issue, let's focus only on SNPs that are fully observed (i.e. the total counts of individuals = all 938 individuals for the full data problem).

Computing and plotting a MAF frequency spectra

```
plot.MAF <- function(file){
  # Read in the HWE table and compute counts and allele frequencies
  hwe<-read.table(file,as.is=TRUE)
  names(hwe)<-c("chr","SNP.id","which.inds","a1","a2","genotype","obs.het","exp.het","HWE.pval")
  counts<-sapply(hwe$genotype,function(x){as.numeric(strsplit(x,"/")[[1]])})
  counts<-t(counts)
  tot.counts<-rowSums(counts)
  allele.counts<-(2*counts[,1]+counts[,2])

  # Flip allele counts so that we are sure we always have the minor
  # allele frequency
  # (Note: this uses a trick based on boolean math where true/false = 1/0).
  counts.maf = allele.counts*(allele.counts<=2*tot.counts-allele.counts) + (2*tot.counts-allele.counts)*

  # Set the number of individuals by looking at the sites w/ the most
  # observed data
  n=max(tot.counts)

  # Make the plot but filter on using only sites with fully observed
  # data (i.e. totcounts==n)
  hist(counts.maf[tot.counts==n],
   xlab="Minor allele count",
   ylab="# of SNPs",
   main="Allele frequency spectra",breaks=n)

  # Plot the expected minor allele frequency spectra for the standard
  # neutral model (i.e. constant size population, all loci neutral)
  # To do so we compute, Watterson's estimator of Theta
  S=sum(tot.counts==n & counts.maf>0)
  thetaW=S/sum(1/seq(1,2*n-1))
  # Which determines the expected AFS
  expectedAFS=(1/seq(1,n)+1/(n*2-seq(1,n))) * thetaW
  # And then plot
  lines(seq(1,n),expectedAFS,col=2)
  # Note: This adds a red line displaying the expected AFS shape
  # controlled to match the data w.r.t to Watterson's Theta (i.e. the total number of SNPs).
}
```

Questions

1. The distribution of MAF's does not have the shape you would expect for a constant-sized population. In what ways does it depart from the expectation?
2. What is at least one plausible explanation for the departures? (Hint: This is SNP array data not sequencing data).

Follow-up Activities (Optional)

1. Carry out the same exercise data with a sequencing data set (for example, 1000 Genomes data) or exome chip data.

Admixture

Though, structure within Europe is subtle, we can run the program `admixture` on our set of 6 Euroepan sub-populations. We will use K=6 and see if the method can in fact distinguish the 6 sub-populations.

Running admixture

```
../bin/admixture ../data/H938_Euro.LDprune.bed 6
```

As it runs you will see updates describing the progress of the iterative optimization algorithm. For this data, the program will run for ~100 iterations after the five initial EM steps. If it is taking too long you may want to pull the results file from the `~/shared_data/HGDP_Data/admixture_files` subdirectory.

Plotting the results

When the analysis is finished, you can plot the results in a simple way using the barplot function:

```
# Read in matrix of inferred ancestry coefficients for each individual.
Q=read.table("H938_Euro.LDprune.6.Q")
Qmat=as.matrix(Q)
barplot(t(Qmat),col=c("red","blue","gold","orange","purple","brown"),border=NA,space=0)
```

Or as a better approach, read in the population id's of each individual and plot the individuals sorted by these identfiers:

```

plot.admixture.labeled <- function(clst_file, fam_file, q_file){
  #Read in the Qmatrix
  Q = read.table(q_file)
  Qmat = as.matrix(Q)

  # To be able to label the graph we read in a
  # .clst file with population "cluster" labels for each indiv
  clst = read.table(clst_file)

  # And a fam file from the plink data
  fam = read.table(fam_file)

  # Use the match function to link the family ids with the cluster id
  clst_unord=clst$V3[match(fam$V2,clst$V2)]

  # Re-order alphabetically
  ordered_indices=order(clst_unord)
  QmatO=Qmat[ordered_indices,]

  # Compute where we will place the population labels in the barplot
  n=length(ordered_indices)
  clst_ord=clst_unord[ordered_indices]
  breaks=c(0,which(clst_ord[1:(n-1)]!=clst_ord[2:n]),n)
  nbrks=length(breaks)
  midpts=(breaks[1:(nbrks-1)]+breaks[2:nbrks])/2

  # Make the barplot
  barplot(t(Qmat0),col=c("red","blue","yellow","orange","purple","brown"),border=NA,space=0,inside=TRUE
  abline(v=breaks,lwd=2)
  mtext(levels(clst_ord),side=1,at=midpts,las=2)
}
```

Questions

1. Are individuals from the population isolates (Adygei, Baseque, Orcadian, and Sardinian) inferred to have distinct ancestral populations?
2. Are the French and Italian individuals completely distinguished as being from distinct populations?
3. Which sampled population would seem to have the most internal population structure?

Follow-up Activities (Optional)

1. Run the method with K=4 and K=5 and describe results.
2. Use the worldwide pruned LD data and run with K=6 or K=7 (i.e. revisiting Rosenberg's classic paper). [Note : this will take a while, best to run it on your own computer, or use a different algorithm like `teraStructure`]
3.

PCA

Principal components analysis is a commonly used way to investigate population structure in a sample (though it is also sensitive to close relatedness, batch effects, and long runs of LD, and you should watch for these potential effects in any analysis). Here you will run PCA on the Euroepan subset of the data with the LD pruned data.

Setting up a parameter file and running smartpca

First set-up a basic smartpca parameter file. Use a text editor to store the following into a file `H938_Euro.LDprune.par` (try `pico` if you're unfamiliar with UNIX text editors). This file runs smartpca in its most basic mode (i.e. no automatic outlier removal or adjustments for LD - features which you might want to explore later). Note: You may need to change `../data/` to reflect the real path where your files are.

```
genotypename: ../data/H938_Euro.LDprune.bed
snpname: ../data/H938_Euro.LDprune.bim
indivname: ../data/H938_Euro.LDprune.PCA.fam
snpweightoutname: ./H938_Euro.LDprune.snpeigs
evecoutname: ./H938_Euro.LDprune.eigs
evaloutname: ./H938_Euro.LDprune.eval
phylipoutname: ./H938_Euro.LDprune.fst
numoutevec: 20
numoutlieriter: 0
outlieroutname: ./H938_Euro.LDprune.out
altnormstyle: NO
missingmode: NO
nsnpldregress: 0
noxdata: YES
nomalexhet: YES
```

We need to deal with a pesky smartpca issue that will cause it to ignore individuals in the `.fam` file if they are marked as missing in the phenotypes column.

```
# Deal with pesky smartpca ignore issue by creating new .fam file
awk '{print $1,$2,$3,$4,$5,1}' ../data/H938_Euro.LDprune.fam > ../data/H938_Euro.LDprune.PCA.fam
```

Now run smartpca:

```
smartpca -p H938_Euro.LDprune.par
```

Plotting the results

And make a plot of PC1 vs PC2.

```
# Read in eigenvectors file
PCA=read.table("H938_Euro.LDprune.eigs")
names(PCA)=c("ID",paste("PC",(1:20),sep=""),"CaseControl")

# Note smartpca pushes the plink family and individual ids together so
# we need to extract out the ids afresh; note this code works just for
# this case
ids=substr(PCA$ID,start=6,stop=20)

# Read in clst table and fam file
clst=read.table("../../data/Euro.clst.txt")
# The list of countries as ordered in the fam file
clst_unord=clst$V3[match(ids,clst$V2)]

# Make a blank plot of the right size
plot(PCA$PC2,PCA$PC1,type="n",xlab="PC2",ylab="PC1")
# Add text labels at PC positions with abbreviations of the full
# labels.  The substr function will be used to make automatic abbreviations.
text(PCA$PC2,PCA$PC1,substr(clst_unord,1,2))
```

Make additional plots of PC3 vs PC4, PC5 vs PC6, and PC7 vs PC8. When looking at each plot inspect each axis indepedently to understand what individuals each PC is distinguishing from one another.

Questions (Part A)

1. Are individuals from the population isolates (Adygei, Baseque, Orcadian, and Sardinian) clearly separated by at least one of the top PCs you've plotted?
2. Are the French and Italian individuals completely separated in at least one of the top PCs you've plotted?
3. Do any of the PCs replicate the structure within Sardinia that was inferred in the admixture analysis above?
4. Do the admixture results and PCA results seem to agree with regards to the relationship of the French, Italian, and Orcadian samples?

We have looked at the top 8 PCs, but perhaps we should be looking at more. Plot the proportion of variance explained by each PC (i.e. the value of each eigenvalue normalized by the sum).

Questions (Part B)

1. Based on the proportion of the variation explained - there are a number of PCs that stand out as being more relevant for explaining variation. About how many?

2. From your plots of PC1-PC8 you should see that the lower PCs seem to be picking up individual-level structure, isolating single individuals. At what PC does this first happen?

```
    eval=read.table("H938_Euro.LDprune.eval")
    plot(1:length(eval$V1),eval$V1/sum(eval$V1),xlab="PC",ylab="Eigenvalue")
```

Follow-up Activities (Optional)

1. Read in the `.snpeigs` file and plot the weight of each SNP along the genome for each of the top PCs. If the spatial clustering of the weights is not distributed genome-wide it may indicate a PC is identifying some local genomic structure rather than genome-wide structure. For example, in many European samples, a PC might indentify a common inversion polymorphism on chr 8p23 or 17q.
2. Run PCA on the worldwide pruned LD data and inspect the results.

Demonstration of spurious association due to population structure [Take-Home]

One consequence of population structure is that it can cause spurious associations with phenotypes. In this exercise you will generate a phenotype that has no dependence on genetics - but that does depend on population membership (imagine a trait determined by diet or some other non-genetic factor that varies among populations). You will try to map it and inspect whether the resulting association test p-values are consistent with the null of no genetic effects.

Generate a phenotype [read-only]

First - let's make a file where each individual is assigned a somewhat arbitrary base phenotypic value given by what population they are from (Adygei = 5, Basque = 2, Italian = 5, Sardinian = 0; French = 8; Orcadian = 10)

```
awk '$3=="Adygei"{print $1,$2,5}\
     $3=="Basque"{print $1,$2,2}\
     $3=="Italian"{print $1,$2,5}\
    $3=="Sardinian"{print $1,$2,0}\
    $3=="French"{print $1,$2,8}\
    $3=="Orcadian"{print $1,$2,10}' \
    ../data/Euro.clst.txt > pheno.base.txt
```

Now, using R, let's add some variation around this base value to produce individual-level phenotypes. Note: Nothing genetic about this phenotype!

```
pheno.base=read.table("pheno.base.txt")
# Scale the base phenotype to mean 0, sd 1
pheno.base.scale=scale(pheno.base$V3)
# Add some normally distributed noise (with as much variance as the base phenotype itself already has)
pheno.sim=rnorm(length(pheno.base$V3),mean=pheno.base$V3,sd=1)
# Output the phenotype to a file
write.table(cbind(pheno.base[,1:2],pheno.sim),"pheno.sim.txt",quote=FALSE,row.names=FALSE,col.names=FALS
```

We will use the `pheno.sim.txt` as our phenotype file for mapping.

Map the trait using plink mapping functions

The `--assoc` command in plink will produce p-values for a basic regression of phenotype on additive genotypic score.

```
../bin/plink --bfile ../data/H938_Euro --pheno .../data/pheno.sim.txt --assoc --out H938_Euro_sim.pheno
```

Exploring the results: A Manhattan plot

Read in the plink results contained in the .qassoc output file and make a Manhattan plot of the results.

```
qassoc=read.table("H938_Euro_sim.pheno.qassoc",header=TRUE)

# Make a Manhattan plot
# First set-up a plot of the right size
plot(1:length(qassoc$SNP),type="n",xlab="SNP index",ylab="-log10(p-value)",ylim=c(3,max(-log10(qassoc$P
# Next add the points (note: we only plot points with
# -log10(p-value)>3 to minimze the number of points plotted)
plot.these=which(-log10(qassoc$P)>3)
points(plot.these,-log10(qassoc$P[plot.these]),col=1+qassoc[plot.these,"CHR"]%%2,pch=16)
# Put in a line for a Bonferroni correction (0.05 / length(qassoc$SNP)
abline(h=-log10(0.05/length(qassoc$SNP)),lty=2,col="gray")
```

Questions (Part A)

1. Which chromosomes locations would you be tempted to follow up here?

Inspect a table of the most extreme hits:

```
print(qassoc[head(order(qassoc$P),n=20),])
```

Questions (Part B)

1. The peak on chromosome 6 is near what famous region of the human genome?
2. The peak on chromosome 4 spans the gene for TLR6, a toll-like receptor involved in bacterial recognition that was noted as being highly differentied in Europe (i.e. high FST) by Pickrell et al (2009) in their analysis of this data. Why might a highly differentiated SNP show a stronger signal of spurious association than other SNPs?

Exploring the results: A quantile-quantile plot.

Use the following code in R to make a plot of the observed vs. expected p-values matched by quantile.

```
# Read in the p-values
qassoc=read.table("H938_Euro_sim.pheno.qassoc",header=TRUE)
# Produce expected p-values from the null (i.e. perfectly uniformly
# distributed).
nTests=length(qassoc$SNP)
Unif=seq(1/nTests,1-1/nTests,length=nTests)
# Sort the -log10 p-values (i.e. match on quantile)
logUnifOrder=order(-log10(Unif),decreasing=TRUE)
SNPorder=order(-log10(qassoc$P),decreasing=TRUE)
# Plot the p-values against against each other (Note: we do for only
# the top 150K SNPs to make the number of points plotted smaller)
qmax=max(-log10(qassoc$P),na.rm=TRUE)
plot(-log10(Unif[logUnifOrder][1:150e3]),-log10(qassoc$P[SNPorder][1:150e3]),pch=16,cex=0.5,xlab="-log(
Expected",ylab="-log(p) Observed",,ylim=c(0,qmax));
```

```
# put in a line for the expected relationship (y=x)
abline(0,1);
# Put in a line for a Bonferroni correction (0.05 / length(qassoc$SNP)
abline(h=-log10(0.05/length(qassoc$SNP)),lty=2,col="gray")
```

Questions (Part C)

1. Does there appear to be evidence for a genome-wide inflation of p-values?

Follow-up Activities

1. Simulate p-values from the null uniform distribution and draw a qq-plot.
2. Consider how genomic control be applied in this situation to control population stratficiation.
3. Use the PCs you've computed already (or plink's MDS functions) to rerun the association test controlling for population stratification.