Genetic data

Gibran Hemani

This worksheet will take you through

1. Familiarising yourself with the genetic data format
2. Calculating some summary statistics
3. Performing quality control (QC) to clean the data

A note about PLINK

This is a very versatile tool for data manipulation and analysis. PLINK version 2 is currently being developed (at the time of writing in beta version 1.9), and is markedly faster than the original plink. In this worksheet we will go through some basic commands, but for a full list of possible operations see the website:

https://www.cog-genomics.org/plink2/

Looking at the data

Conceptually, genetic data is stored in matrix form - rows for individuals, columns for SNPs. In practice, this can take many different shapes, styles and conventions. In these practicals we will be using the most common type - PLINK format. More specifically, **binary** plink format. You can find information about it here:

http://pngu.mgh.harvard.edu/~purcell/plink/data.shtml#bed

A binary plink dataset comprises three files:

1. The `data.fam` file - this has information about the individuals
2. The `data.bim` file - this has information about the SNPs
3. The `data.bed` file - this is the matrix of genotypes for all SNPs and individuals. Note that this is not human readable.

Navigate to the `data/genetic` directory

```
cd data/genetic
```

Look at the files there

```
ls -lh
```

Look at the `geno_unclean.fam` file

```
less geno_unclean.fam
```

```
id1 id1 0 0 2 -9
id2 id2 0 0 2 -9
id3 id3 0 0 2 -9
id4 id4 0 0 1 -9
id5 id5 0 0 1 -9
```

```
id6 id6 0 0 2 -9
id7 id7 0 0 2 -9
id8 id8 0 0 1 -9
id9 id9 0 0 1 -9
```

(to exit press `q`). It contains 6 columns:

1. Family ID
2. Individual ID
3. Father ID
4. Mother ID
5. Sex (1=male, 2=female, 0=missing)
6. Phenotype (-9=missing)

The number of lines represents the number of individuals in the data

```
wc -l geno_unclean.fam
```

Look at the `geno_unclean.bim` file

```
less geno_unclean.bim
```

```
1       rs12562034      0       768448  A       G
1       rs9442372       0       1018704 A       G
1       rs3737728       0       1021415 A       G
1       rs6687776       0       1030565 T       C
1       rs9651273       0       1031540 A       G
1       rs4970405       0       1048955 G       A
1       rs12726255      0       1049950 G       A
1       rs11807848      0       1061166 C       T
1       rs9442373       0       1062638 C       A
1       rs2298217       0       1064979 T       C
```

It also contains 6 columns:

1. Chromosome
2. SNP ID
3. Genetic position
4. Physical position
5. Allele 1 (normally the minor allele)
6. Allele 2

The number of lines represents the number of SNPs in the data

```
wc -l geno_unclean.bim
```

How many chromosomes are there? How many SNPs are on each chromosome?

```
cut -f 1 geno_unclean.bim | uniq -c
```

The `geno_unclean.bed` file is much larger - it contains the compressed matrix of genotypes for each individual at each SNP. It is not human readable, but you can look at a small section by extracting it to a human readable format using plink:

```
../../software/plink_mac \
--bfile geno_unclean \
--chr 22 \
--recode \
--out geno_unclean_chr22
```

Let's take a look at this command.

- `../../software/plink_mac` is calling the plink software
- `--bfile geno_unclean` is saying there is a binary plink file with this prefix (no need to put bed/bim/fam - it will look for those automatically)
- `--chr 22` is saying only keep SNPs on chromosome 22 (just doing this for speed here)
- `--recode` is saying recode the data to a human readable format (known as ped format)
- `--out geno_unclean_chr22` is saying save the recoded ped files with this new prefix

Now have a look at the created files

```
less -S geno_unclean_chr22.ped
```

```
id1 id1 0 0 2 -9 T T A A C C C C T C G G
id2 id2 0 0 2 -9 G T C A C C T C C C G G
id3 id3 0 0 2 -9 T T C A A C C C C C G G
id4 id4 0 0 1 -9 T T C A C C C C T C G G
id5 id5 0 0 1 -9 T T C A C C C C T C G G
id6 id6 0 0 2 -9 T T A A C C C C C C G G
id7 id7 0 0 2 -9 G T C A C C T C T C G G
id8 id8 0 0 1 -9 G G C C C C C C C C G G
id9 id9 0 0 1 -9 T T C A A C C C C C G G
```

Each SNP is now represented by two columns - one column for each allele.

Calculating allele frequencies

The easiest way to calculate the allele frequency of each SNP is to use plink

```
../../software/plink_mac \
--bfile geno_unclean \
--freq \
--out geno_unclean
```

This produces a file called `geno_unclean.frq`

```
less geno_unclean.frq
```

CHR	SNP	A1	A2	MAF	NCHROBS
1	rs12562034	A	G	0.1046	16474
1	rs9442372	A	G	0.4179	16474
1	rs3737728	A	G	0.2755	16474
1	rs6687776	T	C	0.1524	16474
1	rs9651273	A	G	0.2752	16474
1	rs4970405	G	A	0.1028	16474
1	rs12726255	G	A	0.1352	16474
1	rs11807848	C	T	0.4048	16474
1	rs9442373	C	A	0.4415	16474

It has a row for each SNP. We can use R to plot a histogram of the allele frequencies in these data.

Open up R Studio and perform the following commands in the R Studio console

First set the working directory to where the genetic data is. Go to `Session -> Set working directory -> Choose directory....`

Next read in the file

```
frq <- read.table("geno_unclean.frq", header=TRUE)
```

How many SNPs are there?

```
nrow(frq)
```

Now make a histogram of the `MAF` column

```
hist(frq$MAF, breaks=100)
```

Are there any **rare** variants? e.g. MAF < 0.01?

```
table(frq$MAF < 0.01)
```

Note that similar stats can be calculated by plink for missingness rates for SNPs and individuals (using the `--missing` flag), and tests for Hardy Weinberg equilibrium for each SNP (using the `--hardy` flag).

You can now return back to the Terminal

Cleaning the data

We want to filter the data to the following parameters:

- SNPs with more than 5% missing values removed
- Individuals with more than 5% missing values removed
- SNPs with allele frequency < 0.01 removed
- SNPs with Hardy Weinberg disequilibrium p value $<$ 1e-6 removed

We can do all of this in plink like so:

```
../../software/plink_mac \
--bfile geno_unclean \
--maf 0.01 \
--hwe 1e-6 \
--mind 0.05 \
--geno 0.05 \
--make-bed \
--out geno_qc
```

A note on these parameters:

- `--maf 0.01` specifies that SNPs with MAF less than 0.01 should be removed
- `--hwe 1e-6` specifies that SNPs with HW disequilibrium p value < 1e-6 should be removed
- `--mind 0.05` specifies that individuals with more than 5% of SNPs missing should be removed
- `--geno 0.05` specifies that SNPs with more than 5% of individuals with missing data should be removed
- `--make-bed` specifies that once the filtering is done, create a new binary plink data set with the prefix specified with the `--out` flag.

Note that the log from plink records what has been done:

```
less geno_qc.log
```

In this case, a few SNPs were removed due to HWE and MAF.

Removing cryptic relateds

If there is a pair of individuals in the dataset who are related (e.g. cousins, siblings, duplicate samples etc) then we will want to discard one of the pair, so that the final dataset is comprised entirely of unrelated individuals. We will look at how this is done later on, but plink can do this as follows:

```
../../software/plink_mac \
--bfile geno_qc \
--rel-cutoff 0.025 \
--out relateds
```

NOTE THAT THIS MAY TAKE A LONG TIME TO RUN To cancel it press `ctrl+c`.

Assessing population stratification

High density SNP data can be used to cluster individuals based on genetic similarity using principal components analysis (PCA). This is a useful tool to check that all the individuals in your sample come from the same population.

Calculating PCs is sensitive to regions of high linkage disequilibrium. The correct way to calculate PCs in genetic data is to follow these steps:

1. Tdentify a set of LD pruned SNPs
2. Remove regions with high LD
3. Calculate principal components using only these SNPs

This can be done as follows (no need to run this - the PCs have already been created)

```
# pairwise LD pruning
../../software/plink_mac \
--bfile data \
--indep-pairwise 10000 5 0.1 \
--out indep

../../software/plink_mac \
--bfile data \
--extract indep.prune.in \
--make-bed \
--out indep

# remove high ld regions
awk -f ../../scripts/highldregionsb37.awk indep.bim > highldregions.txt

../../software/plink_mac \
--bfile indep \
--exclude highldregions.txt \
--make-bed \
--out indep_ld

# Do PCA
../../software/plink_mac \
--bfile indep_ld \
--pca --out geno_qc
```

NOTE THAT THIS MAY TAKE A LONG TIME TO RUN To cancel it press `ctrl+c`.

The principal components have already been precomputed and they are in the `covs.txt` covariates file. We have also calculated the principal components together with different samples from around the world (HapMap3 data). We will use R to look at these data.

Perform the following commands in the R Studio console

First load up the principal components data

```
load("popstrat.RData")
```

What does the data look like?

```
head(popstrat)
```

It contains IDs, PCs 1-10, and a population identifier. Which populations are present?

```
table(popstrat$population)
```

Our data is there, along with a number of other samples with the following population codes:

> ASW - African ancestry in Southwest USA CEU - Utah residents with Northern and Western European ancestry from the CEPH collection CHB - Han Chinese in Beijing, China CHD - Chinese in Metropolitan Denver, Colorado GIH - Gujarati Indians in Houston, Texas JPT - Japanese in Tokyo, Japan LWK - Luhya in Webuye, Kenya MXL - Mexican ancestry in Los Angeles, California MKK - Maasai in Kinyawa, Kenya TSI - Toscani in Italia YRI - Yoruba in Ibadan, Nigeria

We can try to identify clusters by eye by plotting the principal components against each other. We will use the `ggplot2` library to plot these data

```
library(ggplot2)
ggplot(popstrat, aes(x=PC1, y=PC2)) +
    geom_point(aes(colour=population))
```

A clear clustering can be seen based on population here. To get a better idea of whether or not there are outliers in our data we can re-plot with less confusing colouring:

```
ggplot(popstrat, aes(x=PC1, y=PC2)) +
    geom_point(aes(colour=population=="Our data",
                   alpha=population=="Our data"))
```