University of the Witwatersrand, Johannesburg

Wits Bioinformatics

Population Structure Tutorial

H3A Bionet

2015

1 PCA – plink

1.1. For the first data set, we'll be using the data in *hapmap3*. This is a subset of the HapMap data and it has been cleaned and pruned already

1.2. Run PCA using PLINK

```
plink --bfile hapmap3 --pca --out hapmap3
```

1.3. This performs the PCA and produces several output files

1.4. The most important ones are the *hapmap3.eigenvec* and hapmap3.eigenval files. Inspect them and understand their contents.

1.5. Run Genesis program (*genesis*) to view the output. Use the small.phe file as the phenotype and then play with the Genesis program to view different PCs.

2 Eigenstrat

Eigenstrat is another program for producing PCAs. It is slower, but it also produces much more statistical information. Let's try use it.

2.1. Eigenstrat can take plink-style files as input. It works best if the phentotype column of the fam file contains the population label. I have a helper script to produce this. First fetch the script and then run it – this uses the fifth column of the small.phe file to replace the phenotype in hapmap3.fam with the population label, creating a new file called hapmap3.fam.new (we don't overwrite the old file!)

```
wget www.bioinf.wits.ac.za/software/poputils/popifyfam.py

python popifyfam.py --popfname ../small.phe --popcol 5 \
        --output  hapmap3.fam.new hapmap3.fam
```

2.2. Then we can run smartpca

```
smartpca.perl -i hapmap3.bed -a hapmap3.bim -b hapmap3.fam.new \
              -p hapmap3.pca -e hapmap3.eval -o hapmap3.pca \
              -q NO -l hapmap3-smartpca.log
```

2.3. The eigenvectors can be found at *hapmap3.pca.evec*. The first line contains the eigevalues.

2.4. A PDF file can be found at *hapmap3.pca.pdf*.

2.5. The log file can be found at *hapmap3-smartpca.log*. Have a look at the file and work through it to understand what it has.

2.6. *smartpca.perl* is a wrapper script for the smartpca program. It creates a parameter file called *hapmap3.pca.par*. You may want to change some parameters

- numoutevec: number of eigenvectors produced;
- numoutlieriter, outliersigmathresh: how outliers are dealt with.

You can direcly edit this file and then run smartpca thus

```
smartpca -p hapmap3.pca.par
```

3 Admixture

3.1. Run admixture

```
admixture hapmap3.bed 4
```

3.2. This produces a .Q file that contains estimates for each person and a .P file that contains estimates for each SNP. Look at the data and understand.

3.3. Try different values of K.

3.4. Use Genesis to display the output.

3.5. Since Admixture is stochastic it is a good idea to use CLUMPP to average out several runs of Admixture.

4 Bigger example

We start with the data in *hmpops*. I have cleaned the YRI, LWK and CEU data sets so we'll start by merging them.

4.1. Create a file called *merged.txt* using a text editor.

```
CEU.bed CEU.bim CEU.fam
YRI.bed YRI.bim YRI.fam
LWK.bed LWK.bim LWK.fam
```

4.2. Merge the data

```
plink --merge-list merged.txt --make-bed --out all
```

4.3. We'll also need a phenotype file that gives for each individual their group membership. There are various ways of creating this – you can do it manually. I have a simple script called *fams2phe* that uses the *name* of a fam file as a population label and then the contents of the file to extract the people

```
../fams2phe  YRI.fam CEU.fam LWK.fam  > all.phe
```

Have a look at it

```
head all.phe
```

4.4. You need to prune the data

```
plink --allow-no-sex --bfile hapmap3 --indep-pairwise 50 10 0.2 \
      --out /tmp/hapmap3

plink --allow-no-sex --bfile hapmap3 --extract /tmp/hapmap3.prune.in \
        --make-bed --out hapmap3-prune
```

4.5. Now do a PCA and admixture chart.

4.6. Now also add the other data sets in hmpops. You'll have to do some cleaning to get it to work properly. Remember you should merge then prune, not the other way!