University of the Witwatersrand, Johannesburg

Wits Bioinformatics

Population Structure Tutorial

South African Human Genome Programme

July 2014

1 Setup

1. In this exercise we use the following programs
   - eigenstrat
   - admixture
   - plink
   - CLUMPP
   - distruct
2. You need to use the following data
   - The ALL.{bed,bim,fam} files.
   - group1.phe, group2.phe
3. Scripts in `http://www.bioinf.wits.ac.za/software/poputils`
4. The Genesis program `http://www.bioinf.wits.ac.za/software/genesis`

2 Pruning

Typically population analysis programs (e.g., structure, PCA, relatedness) require SNPs that are not in LD so that we have random samples.

Use plink for this:

1. Use plink to produce *prune* sets – SNPs in LD, not in LD.
2. Remove them

```
plink --allow-no-sex --bfile xxx --indep-pairwise 50 10 $R2 --out /tmp/xxx
plink --allow-no-sex --bfile xxx --extract /tmp/xxx.prune.in \x
   --make-bed  --out xxx-prune
```

See the *do-prune.sh* script.

3 PCA – Eigenstrat

1. Prune the ALL data set
2. Run Eigenstrat by running my wrapper.

   ```
   runpca ALL-prune.bed
   ```

3. This performs the PCA and produces several output files
4. The most important one is the file with the ALL.pca.evec file.
   (a) The first line, commented. Is the eigenvalues of the corresponding eigenvectors. Essentially these show the relative weightings of the corresponding principal components.
   (b) The remaining lines give, for each, individual their eigenvector, which we can interpret as a point in a high-dimensional space. Typically, we select a few of the dimensions for display. Commonly we look at PC 1 and PC 2, which are the first two columns but it may be necessary to display others to see more subtle population structure
5. Use Genesis to show the picture

4 Admixture

1. Make a directory 0 and enter into it
2. Run admixture

   ```
   admixture ../ALL.bed 4
   ```

3. 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.
4. Now at the same hierarchy as the directory, create directories 1 and 2 and repeat: however, use the command

   ```
   admixture ../ALL.bed -s time  4
   ```

   This tells admixture to change the random seed (using the current time)
5. Examine the output from the various runs.
6. Now we run CLUMPP
7. CLUMPP takes one parameter paramfile. The important values are
   (a) DATATYPE should be 0
   (b) INDFILE the input file – should contain output of several admixture calls.
   (c) OUTFILE output file
   (d) MISCFILE log file – output file
   (e) K the number of clusters

(f) C the number of individuals

(g) R number of runs (number of data sets in the INDFILE)

(h) M : method to be used 1, 2 or 3. From most accurate and expensive to least accurate and cheaptest.

(i) GREEDY_OPTION If you use 1 or 2 above, choose 2.

8. Key inputfile contains many runs of *admixture*
9. Run admixture four times – after each run, add the Q file to another file.
10. Set up the paramfile – use the given one as a template but you must make changes.

    `clumpp my-paramfile`
11. Use Genesis to draw pictures

5 Other...

5.1 The cdg.py script

1. A manual is available: `http://www.wits.ac.za/software/poputils`
2. A helper script that creates the necessary things for CLUMPP and distruct.

   ```
   python cdg.py -K 4 --glob "[0-2]/" --popfname group1.phe
          --par_clumpp --doclumpp ALL
   ```

   This says look in the directories 0, 1, 2 for files named ALL.4.Q and use those as input to clumpp, and use the group1.phe file as population description. Typically you might have 100 runs.
3. The key output file is ALL.outfile which summarises result.

6 Genesis

Interactive program for creating PCA and Admixture charts.

- `http://www.bioinf.wits.ac.za/software/genesis`

6.1 evec2gp.py for PCA

1. Requires gnuplot

   `python evec2gp.py --phe group1.phe ALL.pca.evec`

   This takes the vectors produced which give "objective" evidence of population structure plus evidence we give based on ascribed membership.

   It produces a gnuplot program.

   `gnuplot ALL.gp`

2. To produce a Postscript and PDF files, you do the following

```
python evec2gp.py --gp-term postscript --epstopdf --gp-output ALL.eps --phe group1.phe A
gnuplot ALL.gp
```

6.2 DISTRUCT

1. DISTRUCT takes output from CLUMPP (or other programm) and some auxiliary files and creates very high quality pictures. It takes one file as input *drawparams*. These are the key values to change

   (a) INFILE_POPQ – for each (external) population average ancestry from ancestral populations

   (b) INFILE_INDIVQ – describes each individual

   (c) INFILE_LABEL_BELOW – a file that contains a list like this

   ```
   5 YRI
   7 LWK
   9 MKK
   12 SABS
   2 HADZA
   13 CHB
   4 CHD
   10 SAN
   1 IMM
   3 CEU
   11 SANDAWE
   6 JPT
   8 GIH
   ```

   (d) INFILE_CLUST_PERM a file with the colours you want to use

   ```
   1 orange
   2 blue
   3 yellow
   4 green
   5 light_purple
   ```

   (e) OUTFILE the name of the output file (it's a PostScript file)

   (f) K number of clusters

   (g) NUMPOPS number of pre-defined populations

   (h) NUMINDS number of individuals

2. cdg.py can help

```
python cdg.py -K 4 --glob "[0-2]/" --popfname group1.phe \
       --par_distruct --dodistruct --outputps ALL-admixture ALL
```