Population genetics summer course, Denmark

Clustering individuals and inferring ancestry with ChromoPainter and fineSTRUCTURE

Garrett Hellenthal, University College London

22/08/2024

For this practical, we will be applying the statistical software `ChromoPainter` and `fineSTRUCTURE` to cluster (real and simulated) individuals based on genetic similarity. We will also be using the programs `GLOBETROTTER` and `SOURCEFIND` to infer individuals' ancestry using `ChromoPainter` output. We will be using a dataset explored in Hellenthal et al 2014, which is freely available and consists of data from the Human Genome Diversity Panel (http://www.cephb.fr/hgdp/) and other resources. The SNPs were ascertained using Illumina chip technology. For this practical, we will work only with chromosome 22, which has 6,812 SNPs.

For this practical, we will further only use the following populations:

Population	**Country**	**Region**	**number of individuals**
Balochi	Pakistan	Central South Asia	21
BantuKenya	Kenya	Africa	11
BantuSouthAfrica	South Africa	Africa	8
Burusho	Pakistan	Central South Asia	25
English	Britain	Europe	6
HanNchina	China	East Asia	10
Kalash	Pakistan	Central South Asia	23
Makrani	Pakistan	Central South Asia	22
Mandenka	Senegal	Africa	22
MbutiPygmy	Congo	Africa	13
Mongola	Mongolia	East Asia	10
NorthItalian	Italy	Europe	12
Orcadian	Britain	Europe	15
Pathan	Pakistan	Central South Asia	22
Sardinian	Italy	Europe	28
Tuscan	Italy	Europe	8
Total			256

I've also added to these a simulated "population" consisting of 20 individuals simulated as descendents of an admixture event occurring 30 generations ago, where 80% of the DNA was contributed from present-day Brahui individuals (from Pakistan, Central South Asia) and the remaining 20% from present-day Yoruba individuals (from Nigeria, Africa). This simulation is from Hellenthal et al 2014 (see Figure 1) and is the example file included with `ChromoPainter`. The populations in the above table will be used as potential ancestry "surrogates" to detect and describe this admixture event.

1 Clustering individuals: CHROMOPAINTER and fineSTRUCTURE

First we will apply `ChromoPainter` and `fineSTRUCTURE` to cluster individuals. For simplicity, we will only cluster based on chromosome 22 data.

Navigate to the folder `FineStructureFiles/`. Extract `ChromoPainterv2` and `fineSTRUCTURE`:

```
tar -xzvf ChromoPainterv2.tar.gz
unzip fs_4.0.0.zip
```

We will use the pre-compiled binary `fs_linux_glibc2.3` in the directory `fs_4.0.0/`. Compile `ChromoPainterv2` with:

```
gcc -o ChromoPainterv2 ChromoPainterv2.c -lm -lz
```

We aim to cluster all individuals in the above table. To do so, we first use `ChromoPainter` to paint each individual from these populations against the others:

```
./ChromoPainterv2 -g example/BrahuiYorubaSimulationChrom22.haplotypes
-r example/BrahuiYorubaSimulationChrom22.recomrates
-t example/BrahuiYorubaSimulation.idfile.txt
-f BrahuiYorubaSimulationSurrogatesOnly.poplist.txt 0 0
-o example/BrahuiYorubaSimulationSurrogatesPaintingChrom22
-a 0 0 -s 0
```

(As mentioned in the lecture, note that you could initially do E-M steps to infer the "switch" (-n) and "mutation" (-M) parameters, but we will instead use default values. In most applications, skipping this E-M step will not make much or any difference, but it is good practice!).

A problem – it may be too slow for this practical, as it takes ≈10min. Therefore I have already done this painting for you, in
`data/BrahuiYorubaSimulationSurrogatesPaintingChrom22....`

Next we will run `fineSTRUCTURE` to cluster individuals based on the
`data/BrahuiYorubaSimulationSurrogatesPaintingChrom22.chunkcounts.txt` output file. This file gives the total number of haplotype segments ("chunks") that each recipient individual copies from each donor individual. To do so, we first need to calculate a nuisance parameter "c", using:

```
Rscript calcC_Continents.R
data/BrahuiYorubaSimulationSurrogatesPaintingChrom22
```

The value printed to screen is 0.173108247367689. We use this value when running `finestructure`:

```
fs_4.0.0/fs_linux_glibc2.3 finestructure -I 1 -c 0.173108247367689 -x
10000 -y 20000 -z 100
data/BrahuiYorubaSimulationSurrogatesPaintingChrom22.chunkcounts.out
BrahuiYorubaSimulationSurrogatesPaintingChrom22.finestructure.out
```

(Note that in real applications, you should probably have each of `‘‘-x’’`, `‘‘-y’’`, `‘‘-z’’` a factor of 100 higher.) To generate a tree using this output, type:

```
fs_4.0.0/fs_linux_glibc2.3 finestructure -c 0.173108247367689 -x 10000 -k
2 -m T -t 1000000
data/BrahuiYorubaSimulationSurrogatesPaintingChrom22.chunkcounts.out
BrahuiYorubaSimulationSurrogatesPaintingChrom22.finestructure.out
BrahuiYorubaSimulationSurrogatesPaintingChrom22.finestructureTREE.out
```

(Note that in real applications you should probably have `‘‘-x’’` a factor of 10 higher.)

We will also make a “coincidence matrix” that gives the proportion of MCMC samples for which each pair of individuals is clustered together:

```
fs_4.0.0/fs_linux_glibc2.3 finestructure -c 0.173108247367689 -e
meancoincidence
data/BrahuiYorubaSimulationSurrogatesPaintingChrom22.chunkcounts.out
BrahuiYorubaSimulationSurrogatesPaintingChrom22.finestructure.out
BrahuiYorubaSimulationSurrogatesPaintingChrom22.finestructureCOINCIDENCE.out
```

A good way to assess whether you have done enough MCMC samples is to run fineSTRUCTURE again, using a different seed (e.g. with “-s 2”):

```
fs_4.0.0/fs_linux_glibc2.3 finestructure -s 2 -I 1 -c 0.173108247367689 -x
10000 -y 20000 -z 100
data/BrahuiYorubaSimulationSurrogatesPaintingChrom22.chunkcounts.out
BrahuiYorubaSimulationSurrogatesPaintingChrom22.finestructureSEED2.out
```

```
fs_4.0.0/fs_linux_glibc2.3 finestructure -c 0.173108247367689 -x 10000 -k
2 -m T -t 1000000
data/BrahuiYorubaSimulationSurrogatesPaintingChrom22.chunkcounts.out
BrahuiYorubaSimulationSurrogatesPaintingChrom22.finestructureSEED2.out
BrahuiYorubaSimulationSurrogatesPaintingChrom22.finestructureSEED2TREE.out
```

```
fs_4.0.0/fs_linux_glibc2.3 finestructure -c 0.173108247367689 -e
meancoincidence
data/BrahuiYorubaSimulationSurrogatesPaintingChrom22.chunkcounts.out
BrahuiYorubaSimulationSurrogatesPaintingChrom22.finestructureSEED2.out
BrahuiYorubaSimulationSurrogatesPaintingChrom22.finestructureSEED2COINCIDENCE.out
```

Finally we will plot some results using R scripts I have provided. Use `CHROMOPAINTERHeatMapPlot.R` to plot a heatmap of the CHROMOPAINTER `chunkcounts.out` output, with individuals clustered according to the results of the initial `fineSTRUCTURE` run:

```
R CMD BATCH CHROMOPAINTERHeatMapPlot.R
```

This will make a new file called `BrahuiYorubaSimulationSurrogatesPaintingChrom22HEATMAPWithTree.pdf`, which contains a heatmap giving the total number of "chunks" (haplotype segments) that each recipient individual (column) copies from each donor individual (row). The tick marks along each axis color individuals based on their population labels (see legend at bottom).

Use `FineStructureCoincidenceMatrixVisualize2Seeds.R` to plot the coincidence matrix for both fineSTRUCTURE runs:

```
R CMD BATCH FineStructureCoincidenceMatrixVisualize2Seeds.R
```

This will make a new file called `BrahuiYorubaSimulationSurrogatesPaintingChrom22FSCoincidencePlot.pdf`, which contains a heatmap giving the proportion of MCMC samples that each individual (rows) is clustered with every other individual (columns). The top left and bottom right triangles give these proportions for the first and second `finestructure` runs, respectively. Individuals are ordered along the axes according to the inferred `finestructure` tree from the first run, i.e. ordered as in the CHROMOPAINTER heatmap.

Use these plots to answer the following questions:

1. Which groups are copied (painted from) least by the other groups? It looks as if African populations (BantuKenya, BantuSouthAfrica, Mandenka) are copied least (note the yellow streaks going left-to-right for these populations). Also, Kalash does not copy much from others (note the vertical yellow streak for Kalash), indicating it is an isolated population.

2. Which groups copy the most from each other? The African populations appear to copy a lot from each other (dark colors), and note that the East Asian populations (HanNChina, Mongola) copy a lot from each other.

3. Do the inferred clusters seem sensible? Yes – all African individuals cluster together, the Kalash cluster together, and East Asians clearly cluster together, etc.

4. Does the inferred tree seem sensible? Yes, vaguely – happily the African populations merge together before merging with non-Africans, for example.

5. How consistent do results from the two runs appear to be? Fairly consistent – Individuals that are uncertain (e.g. in CentralSouthAsia) are uncertain in both fineSTRUCTURE runs, meaning that they are sometimes clustered together and sometimes apart (i.e. non-black colors on `BrahuiYorubaSimulationSurrogatePaintingChrom22FSCoincidencePlot.pdf`).

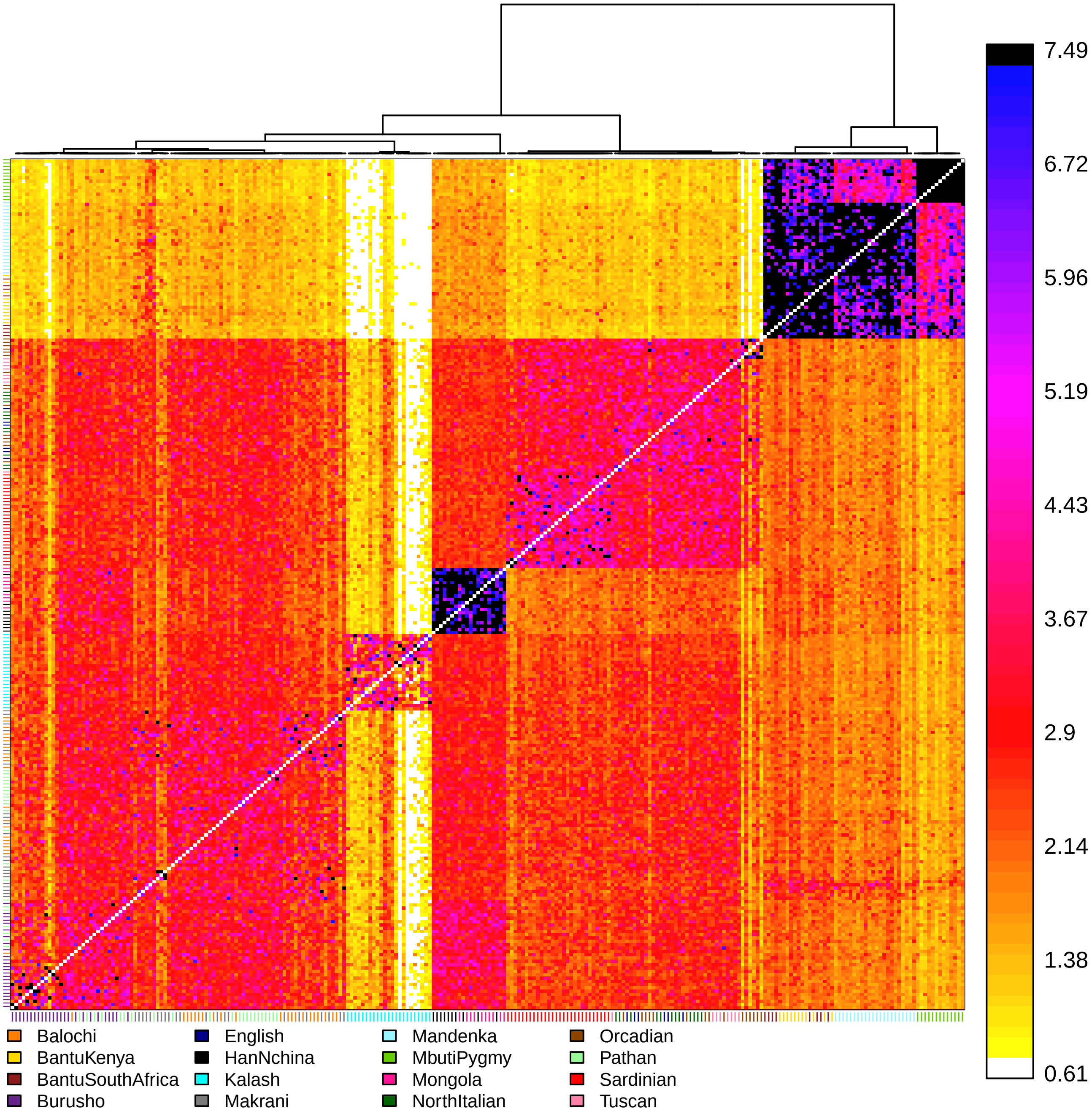

Figure 1: CHROMOPAINTER heatmap, showing the amount of DNA by which each recipient individual (columns) is painted by each donor individual (rows), with inds now ordered by the inferred fineSTRUCTURE clusters (inferred tree at top).

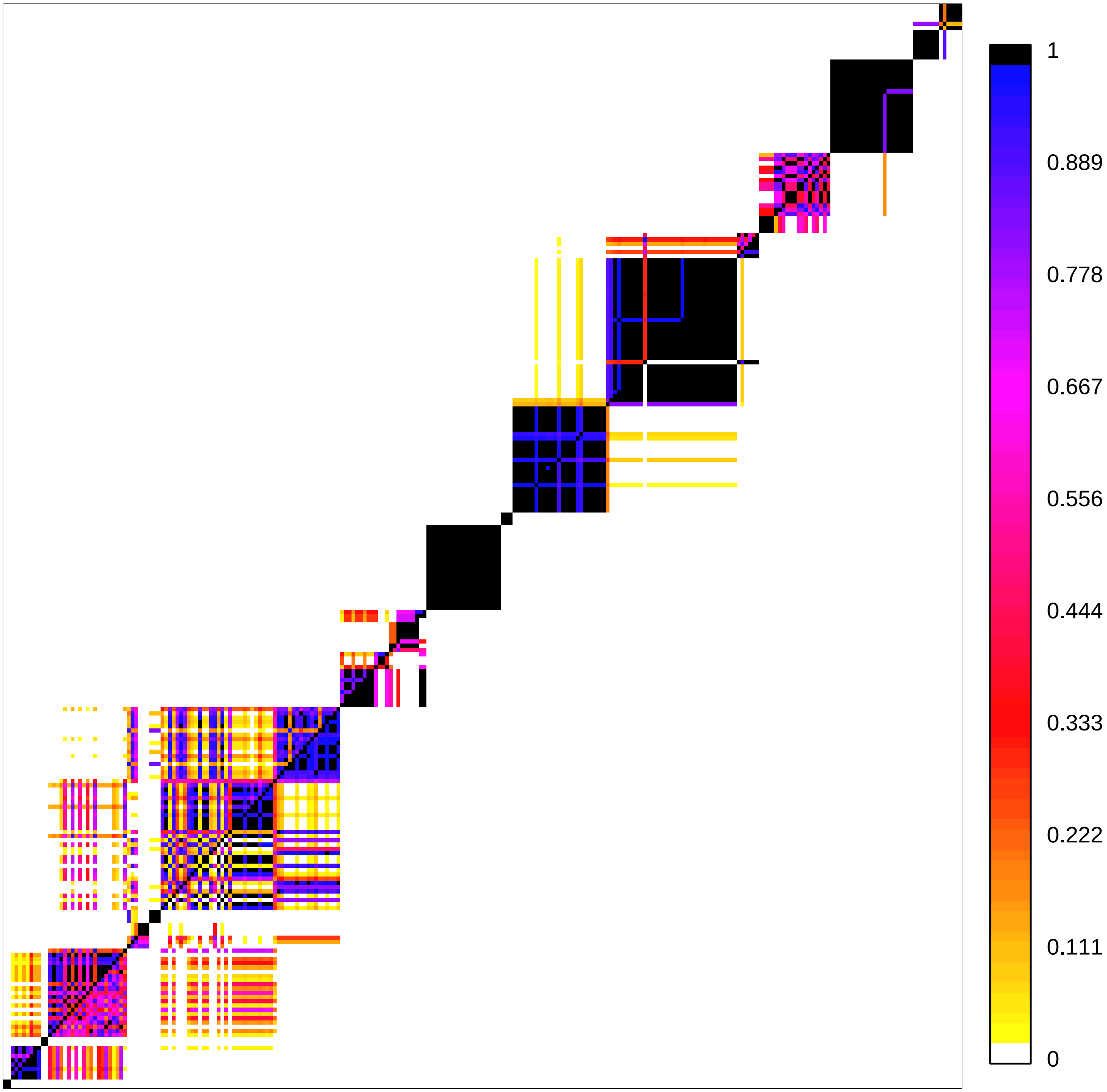

Figure 2: FineSTRUCTURE coincidence matrix, showing the proportion of MCMC samples for which each pair of individuals are clustered together (upper left = fineSTRUCTURE run 1, lower right = fineSTRUCTURE run 2). Individuals are ordered according to the fineSTRUCTURE clusters in Figure 1.

2 Inferring ancestry: GLOBETROTTER and SOURCEFIND

Next we will use GLOBETROTTER and SOURCEFIND to infer ancestry proportions for the simulated population. This will make use of the painting of the ancestry surrogate populations that we did in the previous section. We first need to paint the simulated target individuals against these surrogate populations:

```
./ChromoPainterv2 -g example/BrahuiYorubaSimulationChrom22.haplotypes
-r example/BrahuiYorubaSimulationChrom22.recomrates
-t example/BrahuiYorubaSimulation.idfile.txt
-f BrahuiYorubaSimulation.poplistReduced.txt 0 0
-o example/BrahuiYorubaSimulationAdmixtureChrom22 -s 10
```

The output file of interest here is `example/BrahuiYorubaSimulationAdmixtureChrom22.chunklengths.out`, which gives the total (cM) amount of DNA across chromosome 22 that a target individual copies from each donor poplation. We will combine this painting with that of the surrogates, using a script I made:

```
R CMD BATCH CHROMOPAINTERSurrogateTargetPaintingsCombine.R
```

Next unzip GLOBETROTTER:

```
tar -xzvf GLOBETROTTER.tar.gz
```

and compile with:

```
R CMD SHLIB -o GLOBETROTTERCompanion.so GLOBETROTTERCompanion.c -lz
```

Run GLOBETROTTER using `BrahuiYorubaSimulationAdmixture.paramfileNNLS.txt`, which specifies (using `num.mixing.iterations:0`) that we only want to run the NNLS model in GLOBETROTTER to infer ancestry proportions in the simulated population, and not infer or date admixture:

```
R < GLOBETROTTER.R BrahuiYorubaSimulationAdmixture.paramfileNNLS.txt
--no-save > output.out
```

This will make the output file `example/BrahuiYorubaSimulationAdmixed.GTnnls.main.txt`, which contains the inferred ancestry proportions under the NNLS model.

Now run SOURCEFIND using `BrahuiYorubaSimulationAdmixture.SourcefindParamfile.txt`:

```
tar -xzvf SOURCEFINDv2.tar.gz
```

```
R < sourcefindv2.R
BrahuiYorubaSimulationAdmixture.SourcefindParamfile.txt --no-save >
output.out
```

This will make the output file `BrahuiYorubaSimulation.sourcefind.txt`, which contains the inferred ancestry proportions under SOURCEFIND.

Answer the following questions.

1. How well does the GLOBETROTTER NNLS soluation capture the ancestry of the simulated population? This gives ≈77% Balochi + 12% BantuKenya + 6.5% Mandenka + < 3% contributions from a few other surrogate populations. Thus this gives 77% from Pakistan surrogate groups and 18.5% from SSAfrican surrogate groups, close to the truth.

2. Find the SOURCEFIND MCMC sample with the highest posterior probability. What does this show? How does its inference compare to that of the other MCMC samples (or the mean across samples)? Results will differ, but I get that the SOURCEFIND solution with highest posterior probability infers 70% Balochi + 18% BantuSouthAfrica + 6% Burusho + 3% Sardinian + 3% Pathan. This is slightly noisy, but gives 76% Pakistan groups and 18% SSAfrican, close to the truth. I get that there is a lot of variation across MCMC samples though, and the mean across samples gives 36% Balochi + 24% Markani + 10% Pathan + 6% BantuKenya + 5% Mandenka + 5% BantuSouthAfrica + other groups with contributions ≤3%. This is ≈70% Pakistan groups and 16% SSAfrican. This suggests the MCMC sample with highest posterior probability is better here, and that perhaps more MCMC runs should be used.