Population genetics summer course, Denmark

Dating admixture and detecting selection in admixed individuals

Garrett Hellenthal, University College London

23/08/2024

For the first part of this practical, we will be applying the statistical software `ALDER`, `MALDER`, `fastGLOBETROTTER` and `MOSAIC` to simulated individuals in order to detect and date admixture events.

Here we will use the same dataset as in the "Clustering Algorithms" practical, though now using chromosomes 20, 21 and 22. Again this consists of real data from 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

The aim here is to see how well `ALDER`, `MALDER`, `fastGLOBETROTTER` and `MOSAIC` can reconstruct an admixture event in the simulated "population" described in the "Clustering Algorithms" practical. This simulated group consists of 20 individuals descending from 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). To identify this admixture event, we will use the 16 populations above (or a subset of these populations) as surrogates to the admixing sources.

1 Inferring admixture: ALDER/MALDER

Navigate to the folder `AlderMalderFiles/`. First, we will run ALDER to detect admixture in the simulated population.

Unzip and extract `ALDER`:

```
tar -xzvf alder_v1.03.tar.gz
```

Then compile:

```
cd alder
make
cd ..
```

Then run on the Brahui/Yoruba simulation:

```
alder/./alder -p BrahuiYorubaSimulation.alder.par
> BrahuiYorubaSimulation.alder.out
```

The results of the above run will be in `BrahuiYorubaSimulation.alder.out`.

Also, run `MALDER` on the Brahui/Yoruba simulation. To do so, first unzip and extract `MALDER`:

```
unzip malder-master.zip
```

Then compile:

```
cd malder-master/MALDER/
make
cd ../..
```

Then run on the Brahui/Yoruba simulation:

```
malder-master/MALDER/./malder -p BrahuiYorubaSimulation.malder.par
> BrahuiYorubaSimulation.malder.out
```

Once finished, answer the following questions:

1. Does ALDER detect admixture in this simulation? If so, what is the inferred date?

   The bottom of `BrahuiYorubaSimulation.alder.out` indicates that `ALDER` technically `FAILS` to detect admixture here. This is due to the score when using only `1-ref` (Mandenka) not having a significant admixture signal. However, the `2-ref` results, which use allele frequencies from the surrogates groups to detect admixture, gives a `Z-score` of 7.08, with an inferred date of 32.71 ± 4.62, which is close to the true simulated date. This suggests `ALDER` is conservative in making a call.

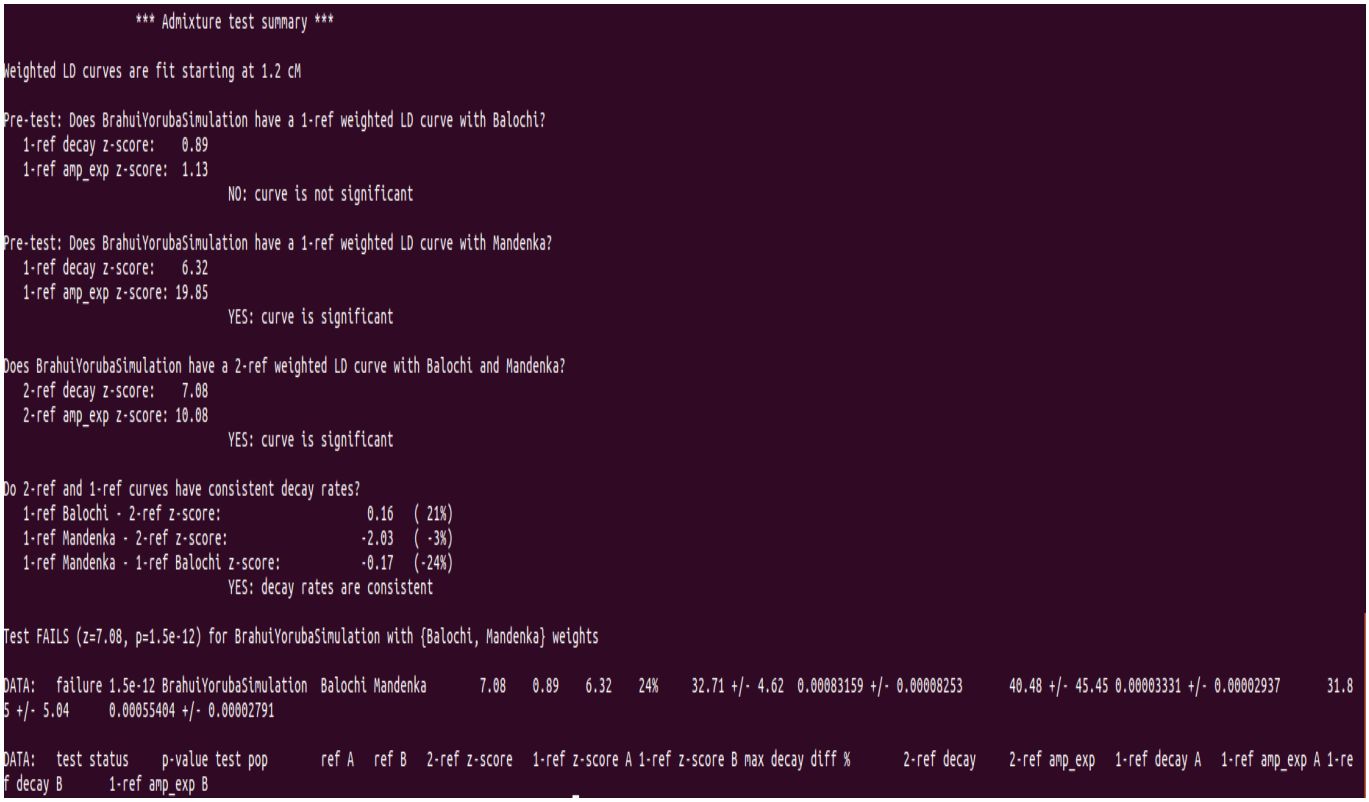


Figure 1: Admixture detection results in `BrahuiYorubaSimulation.alder.out`

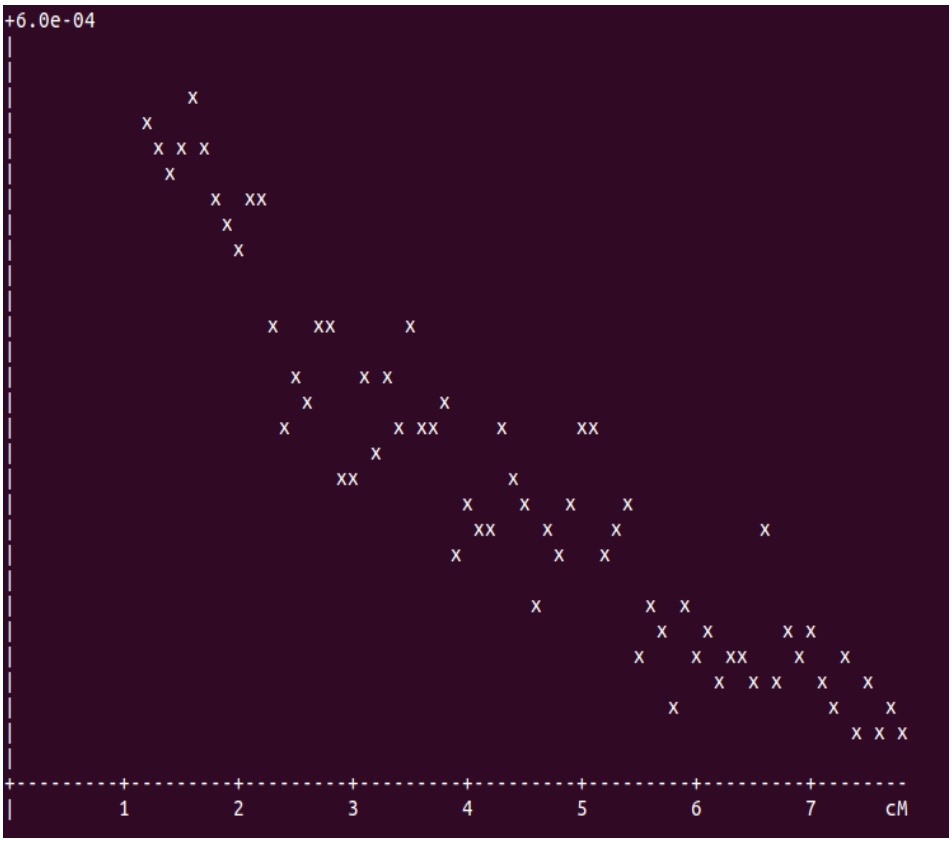
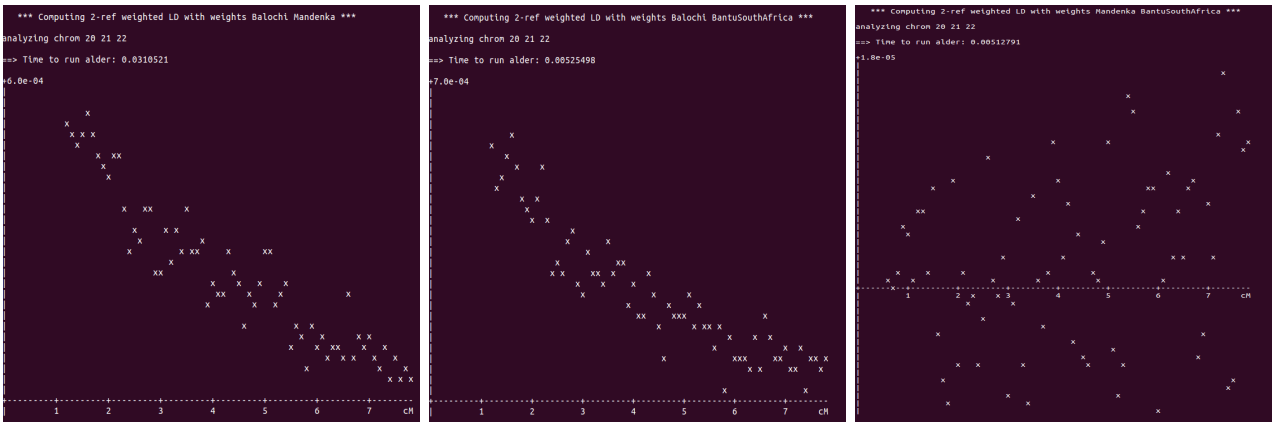


Figure 2: The curve for the Brahui-Yoruba simulation in `BrahuiYorubaSimulation.alder.out`.

2. What does the evidence for admixture look like here?

   The curve depicted in `BrahuiYorubaSimulation.alder.out`, which you can plot yourself using the file `data/BrahuiYorubaSimulation.alder.LDoutput`, shows pretty clear evidence of admixture LD decay.



Balochi vs Mandenka Balochi vs BantuSA Mandenka vs BantuSA

Figure 3: Curve output from `BrahuiYorubaSimulation.malder.out` for different surrogate pair combinations.

3. When running MALDER, does the inferred admixture change when using different combinations of the surrogate populations?

   The output in `BrahuiYorubaSimulation.malder.out` indicates that `MALDER` significantly detects admixture events when using {Balochi, Mandenka} and {Balochi, BantuSouthAfrica} as surrogates, inferring dates of 32.71 ± 2.71 and 34.97 ± 3.47, respectively. But the test fails when using {Mandenka, BantuSouthAfrica} as surrogates. Supporting these conclusions, for the curves depicted in `BrahuiYorubaSimulation.malder.out`, only the first two surrogate pairs show a clear curve, while the {Mandenka, BantuSouthAfrica} curve is extremely noisy. You can plot the curves for the two significant runs yourself using the file `data/BrahuiYorubaSimulation.malder.LDoutput`.

2 Inferring admixture: fastGLOBETROTTER

Navigate to the folder `GlobetrotterFiles/`. As mentioned in the lecture, running `GLOBETROTTER` or `fastGLOBETROTTER` requires three steps:

1. use `CHROMOPAINTER` to paint surrogate populations against each other
2. use `CHROMOPAINTER` to paint target (admixed) populations against surrogates
3. run `GLOBETROTTER` or `fastGLOBTROTTER` using combined results from (1)-(2)

For steps (1)-(2), we will use `ChromoPainterv2`. Unzip and compile `ChromoPainterv2`:

```
tar -xzvf ChromoPainterv2.tar.gz
gcc -Wall -o ChromoPainterv2 ChromoPainterv2.c -lm -lz
```

We have already done step (1) in the last practical. For step (2), we have also done this in the last practical, but note below I have highlighted how we use `-s 10` here to output painting samples:

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

Repeat the above `ChromoPainterv2` command for chromosomes 20 and 21. As mentioned in the lecture, there are two output files of interest for this analysis:
`example/BrahuiYorubaSimulationAdmixtureChrom22.chunklengths.out`
and `example/BrahuiYorubaSimulationAdmixtureChrom22.samples.out`.

(In real applications, we want to sum the `.chunklengths.out` files across chromosomes, and then combine the output from steps (1) and (2). For simplicity here, we will use the combined matrix we made in the previous practical, which is only for chromosome 22, in `data/BrahuiYorubaSimulationAllVersusAllChrom22.chunklengths.out`.)

Finally, for step (3) we'll run `fastGLOBETROTTER` to infer admixture, using this output from `ChromoPainterv2`. Unzip and extract `fastGLOBETROTTER`:

```
tar -xzvf fastGLOBETROTTER.tar.gz
```

Next compile with:

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

To run `fastGLOBETROTTER` for the Brahui-Yoruba simulation, type:

```
R < fastGLOBETROTTER.R BrahuiYorubaSimulationAdmixture.paramfile.txt
BrahuiYorubaSimulationAdmixture.samplesfile.txt
BrahuiYorubaSimulationAdmixture.recomfile.txt 1 --no-save > output.out
```

It will take a few minutes to complete. You can follow progress by typing:

```
pic output.out
```

Once finished, the following output files will be produced, each in the `example/` directory:

`example/BrahuiYorubaSimulationAdmixed.fastGT.main.txt`

`example/BrahuiYorubaSimulationAdmixed.fastGT.main.pdf`

`example/BrahuiYorubaSimulationAdmixed.fastGT.main_curves.txt`

`example/BrahuiYorubaSimulationAdmixed.fastGT.boot.txt`

Using these files, answer the following questions:

1. From the `fastGLOBETROTTER` user manual, what do the different measures in `BrahuiYorubaSimulationAdmixed.fastGT.main.txt` tell you? In particular what is `fastGLOBTROTTER`'s conclusion about admixture in this application? And what are the inferred sources and dates of the admixture event?

   Results will vary somewhat across runs, but the inferred sources and dates give a "best-guess" conclusion of "one-date", which (according to the manual) means a simple admixture event between two sources at one time (which is correct here). Looking at the "1-DATE FIT EVIDENCE", I get that the inferred date is about 31.6 generations ago. The other interesting lines, given we infer a simple admixture event, is "1-DATE FIT SOURCES, PC1:" which tells you the inferred genetic make-up of each of the two admixing sources. My run concludes that 26% of the DNA comes from one source that is related to "BantuSouthAfrica" (contributing 85.2% of the make-up of this source) and "Mandenka" (contributing 11.8%) (i.e. African populations), while the remaining 74% comes from a source that looks like the "Balochi" from Pakistan. So a fairly accurate representation of the 20%-80% Africa-CentralSouthAsia mix we simulated.

2. How do you interpret the coancestry curves in `BrahuiYorubaSimulationAdmixed.fastGT.main.pdf`? Do the results from `BrahuiYorubaSimulationAdmixed.fastGT.main.txt` make sense in light of these coancestry curves?

   The populations with increasing curves always consist of an African population versus a non-African population. Meanwhile curves with two African populations or two non-African populations are all decreasing. This allows `fastGLOBETROTTER` to infer that the two groups that intermixed were African and non-African (mainly Balochi-like), respectively.

3. How confident are the date estimates?

   For this, you want to look at the `example/BrahuiYorubaSimulationAdmixed.fastGT.boot.txt` output file. For the 20 bootstrap re-samples (you can do more if you like; we usually recommend around 100), the dates range from ≈25-37gen ago or so.

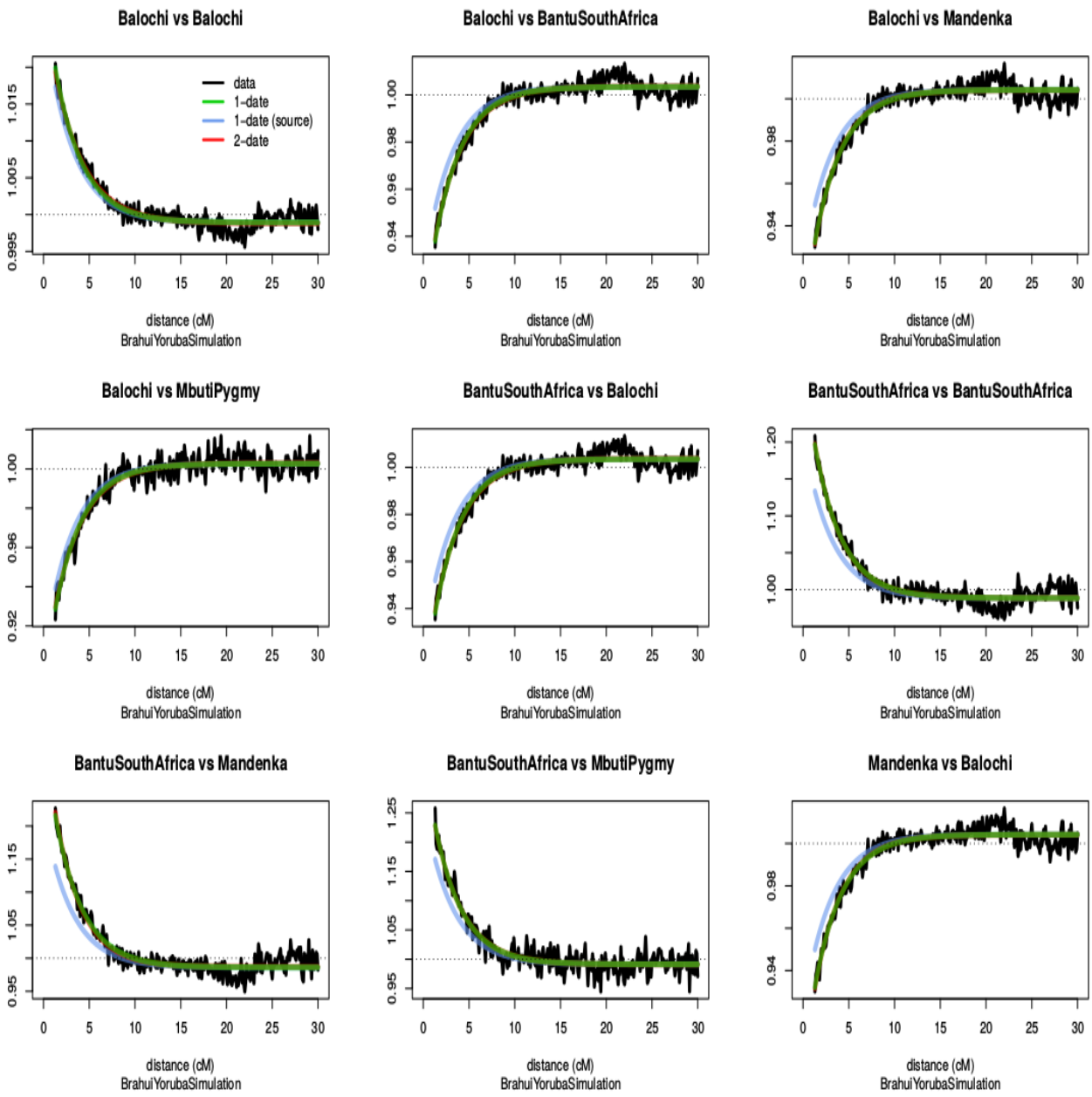


Figure 4: First page of ‘‘example/BrahuiYorubaSimulationAdmixed.fastGT.main.pdf’’

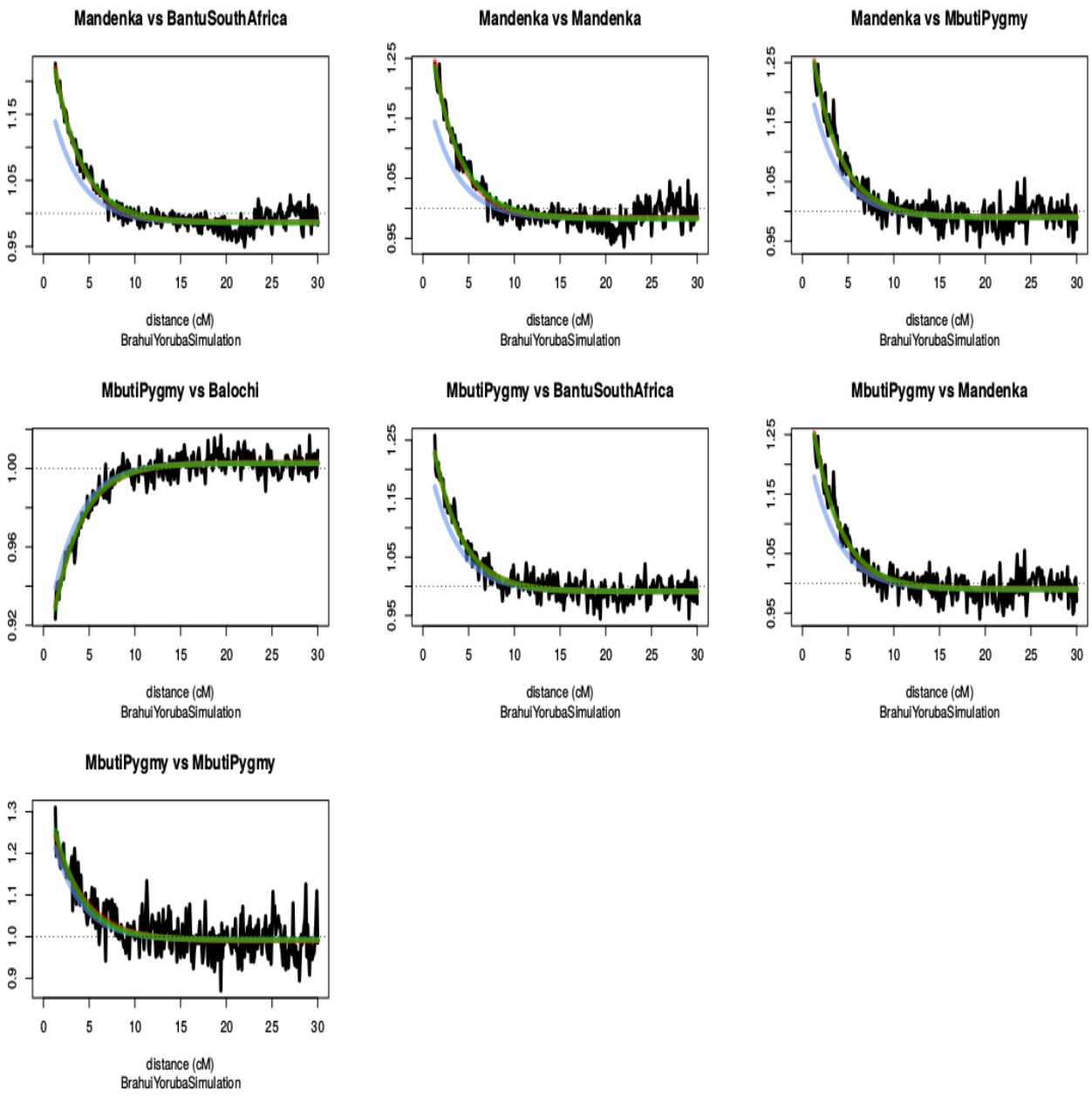


Figure 5: Second page of ‘‘example/BrahuiYorubaSimulationAdmixed.fastGT.main.pdf’’

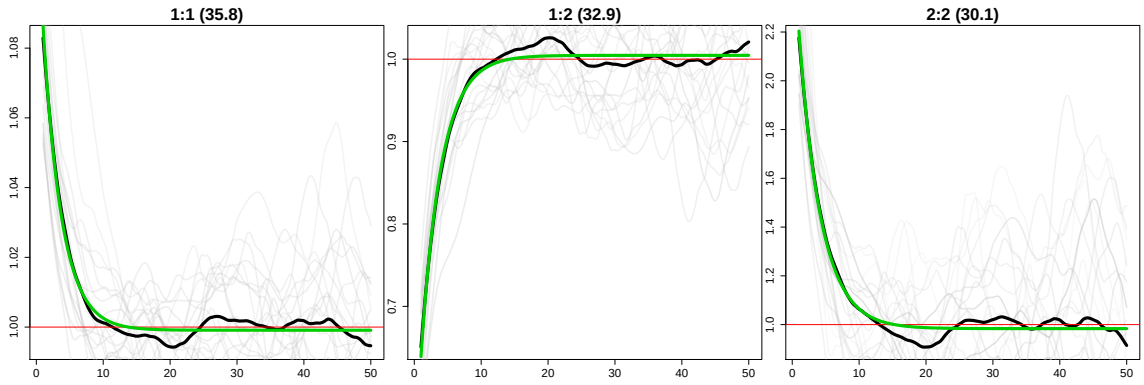


Figure 6: `BrahuiYorubaSimulation_2way_40_20-22_552_60_acoanc.pdf` shows inferred dates for different source combinations.

3 Inferring admixture: `MOSAIC`

Navigate to the folder `MosaicFiles/`. Then unzip and extract `MOSAIC`:

```
tar -xzvf mosaic-master.tar.gz
```

Then run on the Brahui/Yoruba simulation:

```
Rscript mosaic-master/mosaic.R -c 20:22 -p "Balochi BantuKenya BantuSouthAfrica
Burusho English HanNchina Kalash Makrani Mandenka MbutiPygmy Mongola NorthItalian
Orcadian Pathan Sardinian Tuscan" BrahuiYorubaSimulation -a 2 data/
```

It will take a few minutes to complete. The results of the above run will be in three folders: `MOSAIC_RESULTS, MOSAIC_PLOTS, FREQS`).

Looking at the plots in `MOSAIC_PLOTS/`, answer the following questions:

1. What are the conclusions of admixture here, i.e. the inferred date and sources?

   The output files suggest an accurate inferred date ($30.1 - 35.8$ generations) and accurate inferred sources (80% from a Balochi-like source, and 20% from an African-like source).

2. Does it seem as if the algorithm has converged? The `EMlog.pdf` plot shows the likelihood steadily increasing over time, making smaller steps towards the top, indicating convergence to a local maxima.

3. What does the local painting look like?

   The local painting indicates long segments from the Balochi-like and African-like source groups, indicative of a $\approx$30gen admixture date.

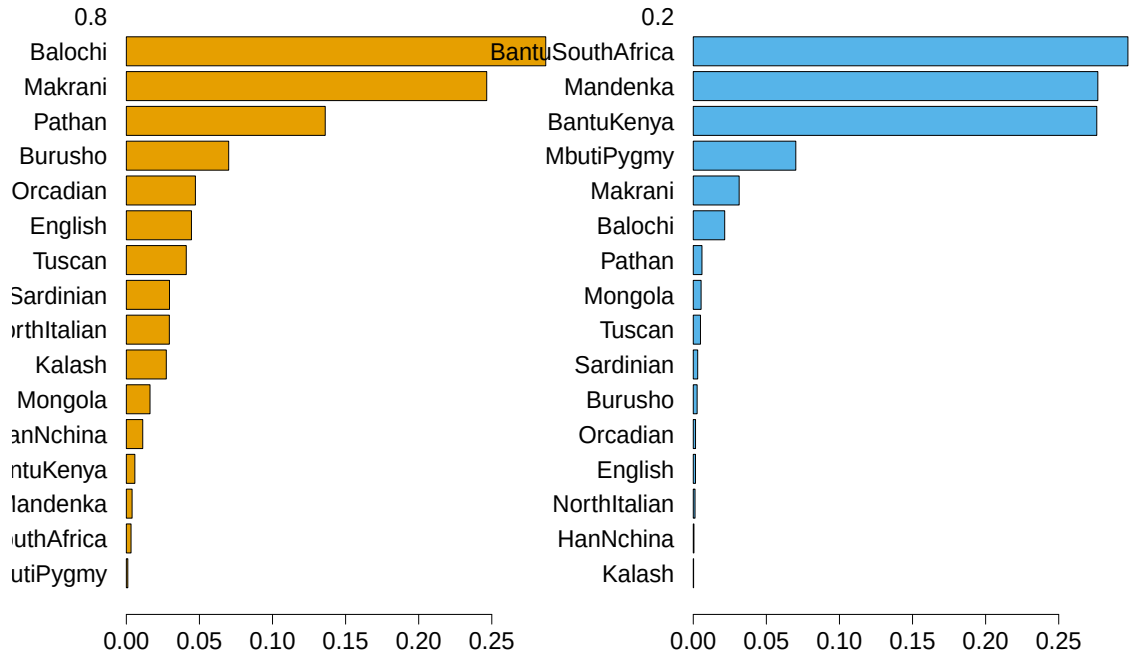


Figure 7: BrahuiYorubaSimulation_2way_40_20-22_552_60_Mu.pdf shows inferred source composition.

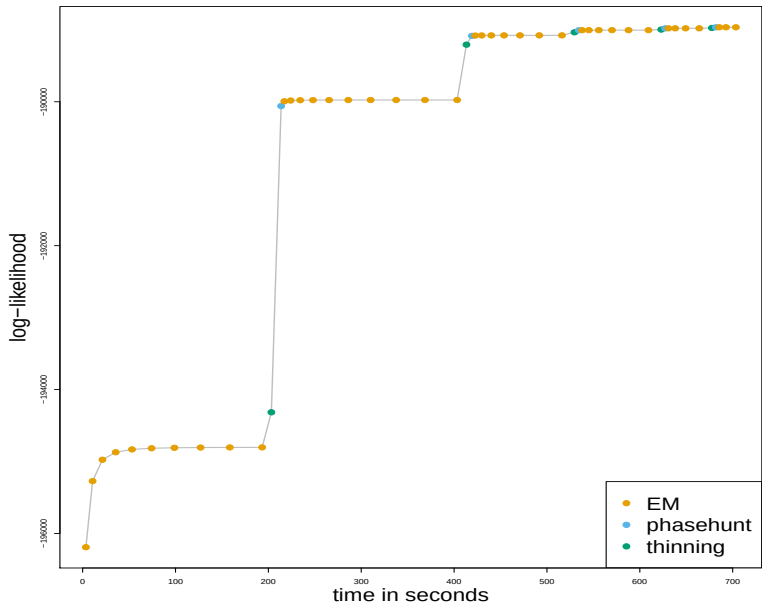


Figure 8: BrahuiYorubaSimulation_2way_40_20-22_552_60_EMlog.pdf indicates the algorithm has converged.

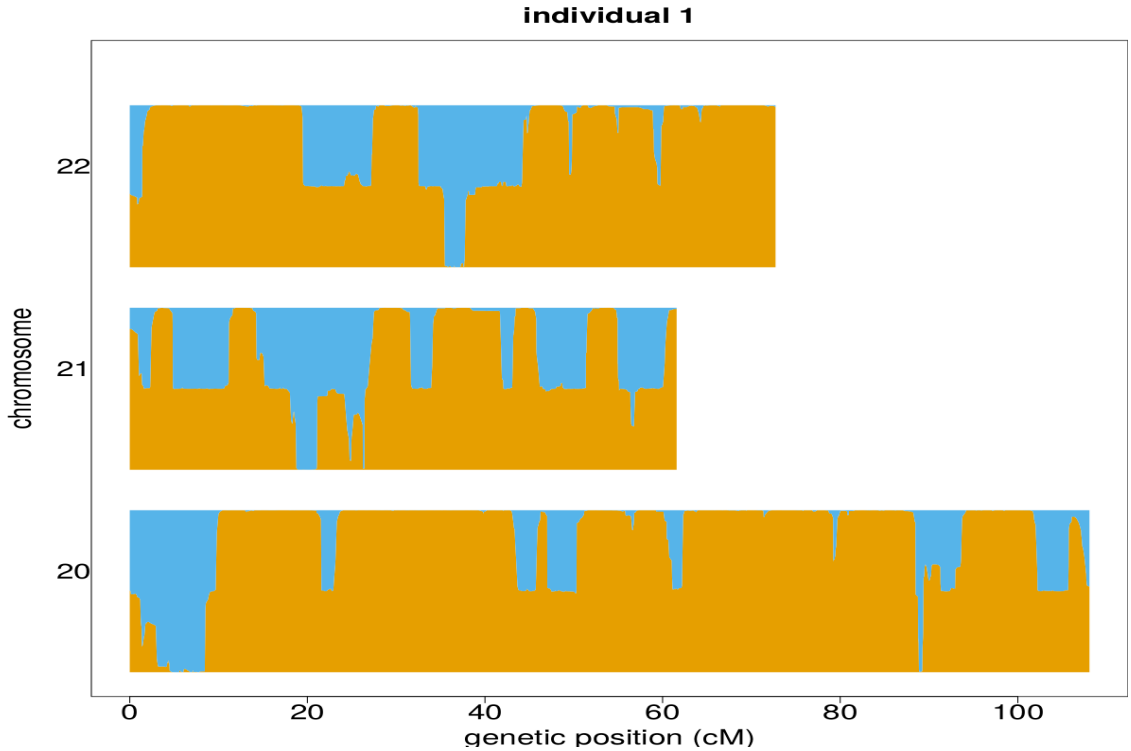


Figure 9: BrahuiYorubaSimulation_2way_40_20-22_552_60_karyograms.pdf for first individual.

4 Inferring selection in admixed inds: ADAPTMIX

For this last section, we will simulate and test for selection in admixed populations with `AdaptMix`, using example data provided with the program. This data is comprised of a small subset of data from 1000 Genomes populations. In particular we will test for selection in a simulated admixed Peruvian population (`PEL`), using admixture surrogates from China (`CHB`), Nigeria (`YRI`) and Spain (`IBS`).

Navigate to the folder `AdaptMixFiles/`. Then unzip and extract `AdaptMix` and `AdaptMixSimulator`:

```
tar -xzvf AdaptMixv1.tar.gz
tar -xzvf AdaptMixSimulator.tar.gz
```

First we will use `AdaptMixSimulator.R`, running it with `CHB_selection_paramfile.txt` and the example data in `simexample/`, to generate a simulated "`PEL`" population that has selection and is admixed from simulated sources related to {`CHB, YRI, IBS`}: (How related the sources are depends on "`drift.btwn.surrogates.and.sources`" in `CHB_selection_paramfile.txt`, with higher values of this making the simulated sources more different from {`CHB, YRI, IBS`}.)

```
R < AdaptMixSimulator.R CHB_selection_paramfile.txt
simexample/PEL_REFs_ALLCHR_chr.txt simexample/PEL_REFs.ids.txt
CHB_selection_ALLCHR --no-save > screenoutput.out
```

This will simulate input data to be read into `run_AdaptMix.R` that consists of the real data for the surrogate populations, added atop a simulated `PEL` population. While nearly all SNPs are neutral, one randomly selected SNP – with starting frequency $\geq$0.05 and $\leq$0.1 (`range.startfrequency.selected.snp:0.05 0.1`) in the population undergoing selection (`CHB`) – will have strong selection (`sel.coeff:0.1` per generation, for 150 generations) occurring *prior* to admixture. This selected SNP will be the last SNP in the output file `CHB_selection_ALLCHR.haps`.

Next run `run_AdaptMix` on this simulated dataset, testing for selection in the simulated `PEL` population:

```
R < run_AdaptMix.R example/PEL_analysis_paramfile.txt
CHB_selection_ALLCHR.txt CHB_selection_ALLCHR.idfile.txt
CHB_selection_ALLCHR.adaptmix.txt --no-save > screenoutput.out2
```

The output will be in `CHB_selection_ALLCHR.adaptmix.txt`, with scores for the selected SNP in the last row of this file. The header is in the third row, with columns giving the p-value of the selection test (column 3) and other information, such as AIC scores.

Repeat this for another simulation described in `PEL_selection_paramfile.txt`, which instead simulates selection post-admixture, with selection strength $s = 0.15$ for 50 generations:

```
R < AdaptMixSimulator.R PEL_selection_paramfile.txt
simexample/PEL_REFs_ALLCHR_chr.txt simexample/PEL_REFs.ids.txt
PEL_selection_ALLCHR --no-save > screenoutput.out

R < run_AdaptMix.R example/PEL_analysis_paramfile.txt
PEL_selection_ALLCHR.txt CHB_selection_ALLCHR.idfile.txt
PEL_selection_ALLCHR.adaptmix.txt --no-save > screenoutput.out2
```

The `AdaptMix` output for this run will be in `PEL_selection_ALLCHR.adaptmix.txt`.

Use the two `AdaptMix` output files for these two simulations to answer the following questions.

1. For each simulation scenario, is there evidence of selection at the SNP with simulated selection? Results will vary per simulation, but I get that there is not much evidence for selection in the first simulation where selection occurs in `CHB` prior to admixture. This is despite the observed frequency in the target population (column 4) being quite different from the expected frequency without selection (column 5). This indicates that power is an issue when trying to assess selection occuring pre-admixture, particularly with the limited sample size (only 85 `PEL` individuals). However, for the second simulation, I do see fairly strong evidence of selection. This difference is likely due to here all individuals experiencing selection, while only individuals with a high amount of `CHB`-related ancestry will show any selection signal in the first simulation.

2. For the SNP with simulated selection in each scenario, do the results indicate selection post-admixture, or in a particular source population pre-admixture? Here you want to use the columns {`AIC.postadmix.target.1, AIC.insurr.source1target.1, AIC.insurr.source2target.1, AIC.insurr.source3target.1`}, which give evidence for selection occurring {post-admixture, pre-admixture in the `CHB`-related source, pre-admixture in the `IBS`-related source, pre-admixture in the `YRI`-related source}, respectively. I get that the lowest AIC score, which indicates the scenario with strongest evidence, is correct in each case.

3. Looking at the bottom of `screenoutput.out`, how well do the correlations between the simulated allele frequencies of the sources and their respective surrogate populations match that observed in the real data? How would you adjust `drift.btwn.surrogates.and.sources` in the input parameter files to make a better match? Here you want to compare `cor.sims`, which represents the former, to `cor.truth`, which represents the latter. In general `cor.sims` seems higher than `cor.truth`, which means you may want to increase the values in `drift.btwn.surrogates.and.sources`. This will then move the allele frequencies of the simulated sources further away from their corresponding surrogates' allele frequencies.