Relate practical: Using genealogies for population genetics

Leo Speidel[1,*], Simon R. Myers[1,2]

[1] Department of Statistics, University of Oxford

[2] Wellcome Centre for Human Genetics, University of Oxford

* Contact: leo.speidel@outlook.com

In this practical, we will look at some real data and analyse the corresponding genealogical trees inferred by Relate. We will use the Simons Genome Diversity Project dataset, downloaded from `https://reichdata.hms.harvard.edu/pub/datasets/sgdp/`. This dataset comprises whole-genome sequencing data of 278 modern humans with sampling locations shown in Fig. 1.

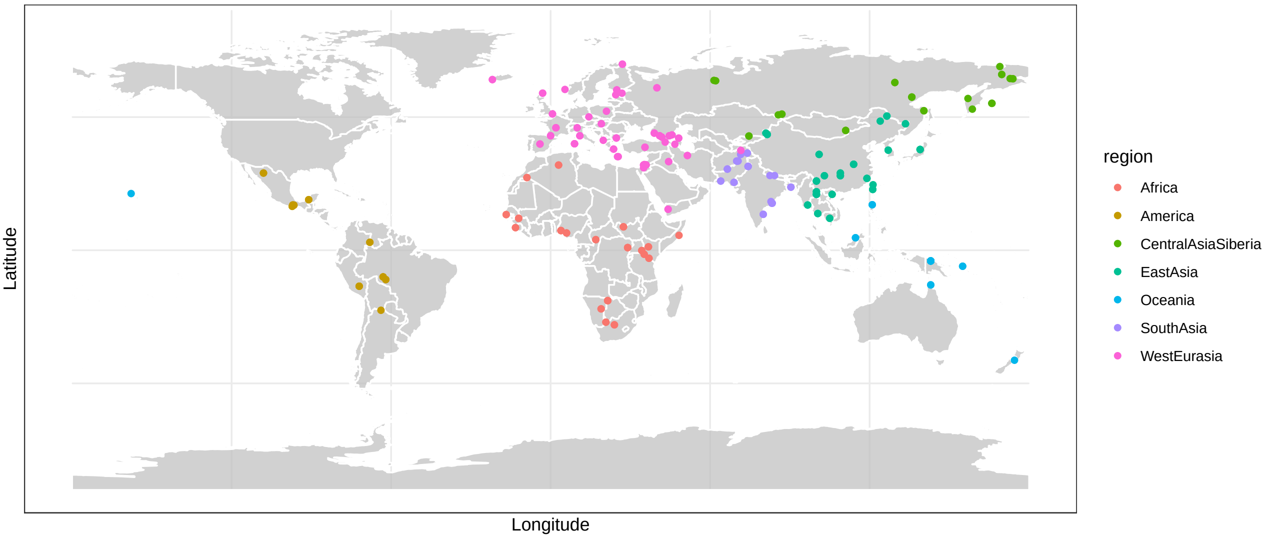

Figure 1: Sampling locations of the 278 modern humans in the Simons Genome Diversity Project. Samples are classified into seven regions shown by colours.

Note 1

The data was downloaded from:

- Phased genotypes: `https://sharehost.hms.harvard.edu/genetics/reich_lab/sgdp/phased_data/PS2_multisample_public`

- Genomic mask: `https://reichdata.hms.harvard.edu/pub/datasets/sgdp/filters/all_samples/`

- Human ancestral genome: `ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase1/analysis_results/supporting/ancestral_alignments/`

- Recombination maps: `https://mathgen.stats.ox.ac.uk/impute/1000GP_Phase3.html`

We then used `RelateFileFormats --mode ConvertFromVcf` to convert to the haps/sample file format and the `PrepareInputFiles.sh` script to make sure ancestral alleles are denoted by 0 and to filter out regions according to the genomic mask.

Overview

1 Relate

Relate estimates the joint genealogies of many thousands of modern individuals genome-wide. These genealogies describe how individuals are related through their most-recent common ancestors back in time and can be seen as the genetic analogue of a family tree for unrelated individuals.

The output of Relate is a sequence of binary trees, each describing the genealogical relationships locally in that part of the genome. Neighbouring genealogical trees differ because of recombination events that change the genetic relationships of individuals.

The method is published in L. Speidel, M. Forest, S. Shi, S. R. Myers. A method for genome-wide genealogy estimation for thousands of samples. Nature Genetics **51**, 1321–1329 (2019).
`https://doi.org/10.1038/s41588-019-0484-x`

Note 2

A detailed documentation for **Relate** is available at `https://myersgroup.github.io/relate`.

For this practical, we will use **bash scripts and R**.
The Relate binaries can be found under `relate_v1.0.17_*/` or downloaded from `https://myersgroup.github.io/relate`. It will be convenient to define a variable storing the location of the Relate binaries, e.g.,

```
PATH_TO_RELATE="/home/popgen/software/relate_v1.0.17_x86_64_dynamic/"
```

1.1 Data requirements and file formats

Relate uses the **haps/sample** file format (output file format of SHAPEIT2) as input
(see `https://myersgroup.github.io/relate/input_data.html#FileFormat`).

- You can convert from **vcf** to haps/sample and from **hap/legend/sample** to haps/sample using functions provided by Relate.

(see code below or `https://myersgroup.github.io/relate/input_data.html#ConvertToHapsSample`)

Note 3: Data requirements

- Please separate your data by chromosome.
- Relate assumes whole-genome sequencing data as input.
- The data needs to be phased. This can be done, for instance, using SHAPEIT.
- Ancestral and derived states at single-nucleotide polymorphisms (SNPs) need to be known.
- A genomic mask should be specified that filters out "bad" regions

We provide a script to prepare your data under

```
${PATH_TO_RELATE}/scripts/PrepareInputFiles/PrepareInputFiles.sh
```

(see `https://myersgroup.github.io/relate/input_data.html#Prepare`).

1.2 The arguments

Required arguments

--mode	Mode in which to run Relate. Mode "All" will execute all stages of the algorithm and other modes can be used to execute individual stages (see here). This is useful, for instance, when parallelizing Relate.
-m,--mutation_rate	Mutation rate per base per generation.
-N,--effective_size	Haploid effective population size. (For diploid organisms, multiply the effective population size of individuals by 2)
--haps	Filename of haps file.
--sample	Filename of sample file.
--map	Filename of genetic recombination map.
-o,--output	Filename of output files without file extension.

Note 4: Optional arguments

There are other optional arguments for Relate, please consult the documentation for these (`https://myersgroup.github.io/relate/getting_started.html#GettingStarted`).

Note 5: Warning

Relate creates temporary files and directories.
Therefore, please do not run more than one instance of Relate in the same directory. (You can run Relate in different subdirectories.)

2 Using Relate on the Simons Genome Diversity Project dataset

Let's start with running Relate. Change into the `workshop_materials/24_relate/` directory

```
cd workshop_materials/24_relate
```

In this directory, you will find three subdirectories, named `data` (containing all input data), `precomp_results` (containing precomputed Relate output files), and `Rscripts` (containing R scripts for plotting results).

2.1 Running Relate

Note 6

Running Relate on all 278 samples and four cores took 52 minutes, so here we will run Relate only on the African samples of this dataset. Precomputed anc/mut files for all samples can be found in the `precomp_results/SGDP_all/` directory.

We run Relate on chromosome 15 for the 44 African samples of the Simons Genome Diversity Project.

```
mkdir results
cd results

chr=15
#This takes about 3 minutes
${PATH_TO_RELATE}/bin/Relate \
     --mode All \
     --haps "../data/data_subgroups/SGDP_input_Africa_chr${chr}.haps.gz" \
     --sample "../data/data_subgroups/SGDP_input_Africa_chr${chr}.sample.gz" \
     --map "../data/genetic_map_chr${chr}_combined_b37.txt" \
     --dist "../data/data_subgroups/SGDP_input_Africa_chr${chr}.dist.gz" \
     -m 1.25e-8 \
     -N 30000 \
     -o "SGDP_Africa_chr${chr}"

gzip SGDP_Africa_chr15.*
```

The haps/sample files store the genetic variation data (similar to a vcf file) and the genetic recombination map stores recombination rates along the genome. In addition, we specify a .dist file to adjust the distances (in units of BP) between SNPs using the genomic mask – this file is outputted by the `PrepareInputFiles.sh` script and is necessary to adjust mutation rates (we can only observe mutations in regions that pass the filters of the mask).

2.2 Extracting trees for subgroups

If we have the genealogies for all 278 samples (precomputed in the `precomp_results/SGDP_all` directory), then we can extract the trees corresponding to a subgroup, e.g., Africans.

```
# in results/ directory
chr=15
pop="Africa"
${PATH_TO_RELATE}/bin/RelateExtract \
    --mode SubTreesForSubpopulation \
    --pop_of_interest ${pop} \
    --poplabels "../data/SGDP_region.poplabels" \
    --anc "../precomp_results/SGDP_all/SGDP_all_chr${chr}.anc.gz" \
    --mut "../precomp_results/SGDP_all/SGDP_all_chr${chr}.mut.gz" \
    -o "SGDP_all_${pop}_chr${chr}"
```

Exercise 1

Let's plot a few trees. We use the TreeView.sh script provided with Relate: For this, we need an additional file storing assignment of individuals to populations. This file has four columns, named "sample", "population", "group", and "sex". For us, only the second column is of interest. The order in which individuals are listed has to be consistent with the order in the samples file.

Example:

```
sample population group sex
UNR1 PJB SAS 1
UNR2 JPT EAS 2
UNR3 GBR EUR 2
UNR4 YRI AFR 2
```

We can now run the TreeView.sh script as follows (this will use R and requires ggplot2 and cowplot - these will be installed if missing):

```
#This will take approx. 2 minutes
${PATH_TO_RELATE}/scripts/TreeView/TreeView.sh \
                 --haps "../data/SGDP_input_chr${chr}.haps.gz" \
                 --sample "../data/SGDP_input_chr${chr}.sample.gz" \
                 --anc "../precomp_results/SGDP_all/SGDP_all_chr${chr}.anc.gz" \
                 --mut "../precomp_results/SGDP_all/SGDP_all_chr${chr}.mut.gz" \
                 --poplabels "../data/SGDP_region.poplabels" \
                 --bp_of_interest 48426484 \
                 --years_per_gen 28 \
                 -o "SGDP_all_chr${chr}_BP48426484"
```

This will produce a pdf named `SGDP_all_chr15_BP48426484.pdf`. Can you see anything unusual about this tree. You can also rerun the above replacing `TreeView.sh` by `TreeViewMutation.sh`; this will highlight branches carrying the mutation at chromosome 15, BP 48426484.

Feel free to plot a few other trees!

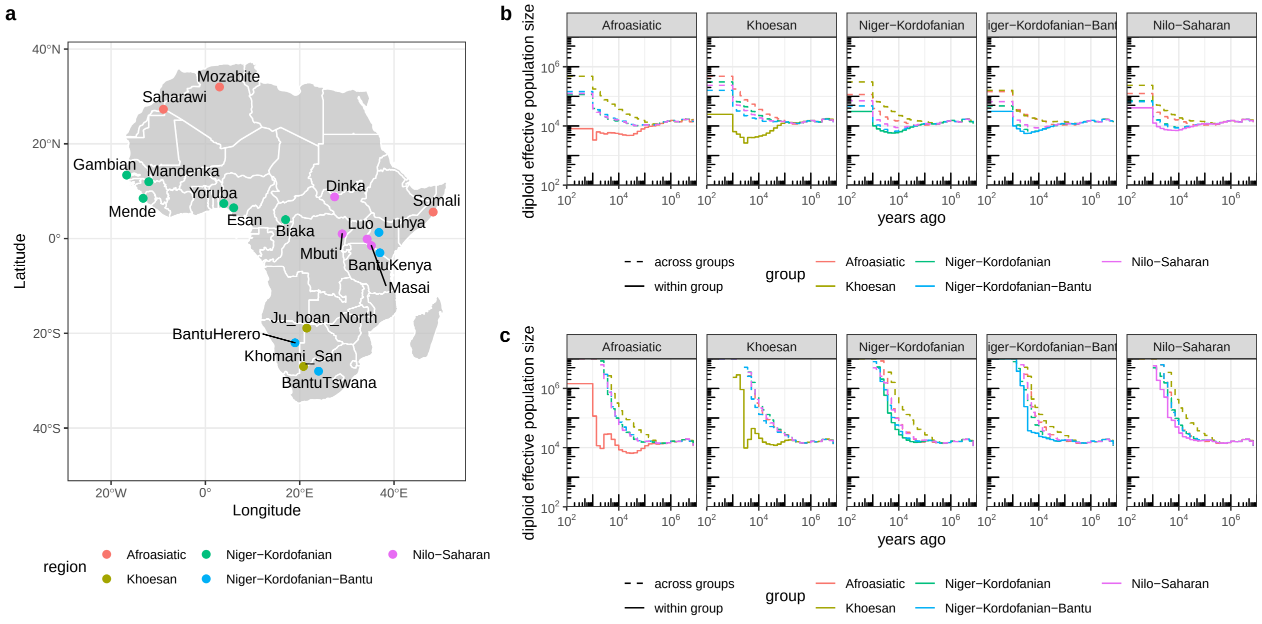

Figure 2: Effective population sizes calculated from the world-wide genealogies estimated by Relate. **a**, Using trees estimated assuming a constant population size of 15,000. **b**, Using trees output after applying the `EstimatePopulationSize.sh` script (see below for details) to jointly fit demographic history and branch lengths.

3 Effective population sizes and split times

Note 7

We will use the genealogies for Africans obtained in Section 2.2 as an example here. You can replicate the analysis for other subgroups, or for all samples; each step will take longer for larger sample sizes (for all samples, computation time in Section 3.1 is around 15 min and for Section 3.2 around 70 min). Precomputed files for the world-wide (all samples) case can be found in the `precomp_results/SGDP_all/` directory.

3.1 Estimating population sizes given a genealogy

We infer effective population sizes and split times from estimated genealogies of Africans obtained in Section 2.2. Extracting effective population sizes from genealogies is done as follows

```
#assuming you are in workshop_materials/24_relate/results
cd ../
mkdir popsizes
cd popsizes

pop="Africa"
chr=15
${PATH_TO_RELATE}/bin/RelateCoalescentRate \
    --mode EstimatePopulationSize \
    -i "../results/SGDP_${pop}_chr${chr}" \
    -o "SGDP_${pop}_chr${chr}" \
    --poplabels "../data/SGDP_${pop}.poplabels"
```

This will produce a file named `SGDP_Africa_chr15.coal` and `SGDP_Africa_chr15.bin`. The `SGDP_Africa_chr15.coal` stores cross-coalescence rates for all pairs of populations in `SGDP_Africa.poplabels` of which there are 19. This might be too many to visualise easily, so we will aggregate some of these groups based on language family. The .bin file can be used to extract coalescence rates for different poplabels files, e.g.

```
${PATH_TO_RELATE}/bin/RelateCoalescentRate \
    --mode FinalizePopulationSize \
    -o "SGDP_Africa_chr${chr}" \
    --poplabels "../data/SGDP_Africa_lang.poplabels"
```

Exercise 2

(i) Plot the estimated effective population sizes stored in `SGDP_Africa_chr15.coal` in R. You can use relater, which is an R package that can parse and manipulate Relate output files (preinstalled, available at `https://github.com/leospeidel/relater`):

```r
library(relater)
coal <- read.coal("SGDP_Africa_chr15.coal")
head(coal)

#diploid effective population size is the 0.5* inverse coalescence rate
coal$popsize <- 0.5/coal$haploid.coalescence.rate

#multiply epochs times by 28 to scale to years (assuming 28 years per generation)
coal$epoch.start <- 28 * coal$epoch.start

#add a column on whether coalescence rate is within or across groups
coal$within <- c("across groups","within group")[(coal$group1 == coal$group2)+1]

#now plot the result
library(ggplot2)
ggplot(coal) +
  geom_step(aes(x = epoch.start, y = popsize, color = group2, linetype = within)) +
  scale_x_continuous(trans = "log10", limit = c(1e2, 1e7)) +
  scale_y_continuous(trans = "log10", limit = c(1e3,1e7)) +
  annotation_logticks(sides = "lb") +
  scale_linetype_manual(values = c(2,1), name = "") +
  facet_grid(.~group1) +
  xlab("years ago") +
  ylab("diploid effective population size")
```

This should give a figure similar to Fig. 2**b**. Can you estimate the split time of Khoesan and other groups? How are the population sizes of groups speaking Afroasiatic languages different from other groups?

(ii) You can also plot the effective population sizes for all samples of the Simons Genome Diversity Project; for this, the relevant precomputed file is located at `../precomp_results/SGDP_chr15.coal`.

3.2 Joint fitting of population size and branch lengths

So far, we used trees that assumed a pre-specified constant effective population size through time. Next, we use the `EstimatePopulationSize.sh` script to jointly fit effective population sizes and branch lengths.

```
chr=15
pop=Africa

${PATH_TO_RELATE}/scripts/EstimatePopulationSize/EstimatePopulationSize.sh \
  -i "../results/SGDP_${pop}_chr${chr}" \
  -o "SGDP_${pop}_ne_chr${chr}" \
  --poplabels "../data/SGDP_${pop}_lang.poplabels" \
  -m 1.25e-8 \
  --num_iter 1 \
  --threshold 0 \
  --years_per_gen 28 \
  --threads 4
```

This will output `SGDP_Africa_ne_chr15.anc.gz`, `SGDP_Africa_ne_chr15.mut.gz`, and `SGDP_Africa_ne_chr15.coal`.

Note 8

Please note that the branch length estimation method assumes one panmictic population – this is clearly not the case for African samples of the Simons Genome Diversity Project. We therefore run this script only for one iteration, to avoid introducing unwanted biases in some populations.

The above script will use four threads (this can be changed). Setting threshold to 0 means that no trees will be removed, otherwise any tree with fewer mutations than this value will be removed.

To run this on more than one chromosome, the `--first_chr` and `--last_chr` arguments can be useful (see documentation
`https://myersgroup.github.io/relate/modules.html#PopulationSizeScript`.)

Exercise 3

(i) Similarly to Exercise 2, estimate the effective population sizes for `SGDP_Africa_ne_chr15.*` and plot these e.g., in R, using relater to load the coal file. This should give a figure similar to Fig. 2**c**. How is this plot different to the one generated in Exercise 2 (which assumed a constant population size through time)? An example script can be found under `Rscripts/`.

(ii) You can repeat this analysis for all samples (which will take longer), precomputed files `SGDP_ne_chr15.bin` and `SGDP_ne_chr15.coal` can be found in the `./precomp_results/` directory.

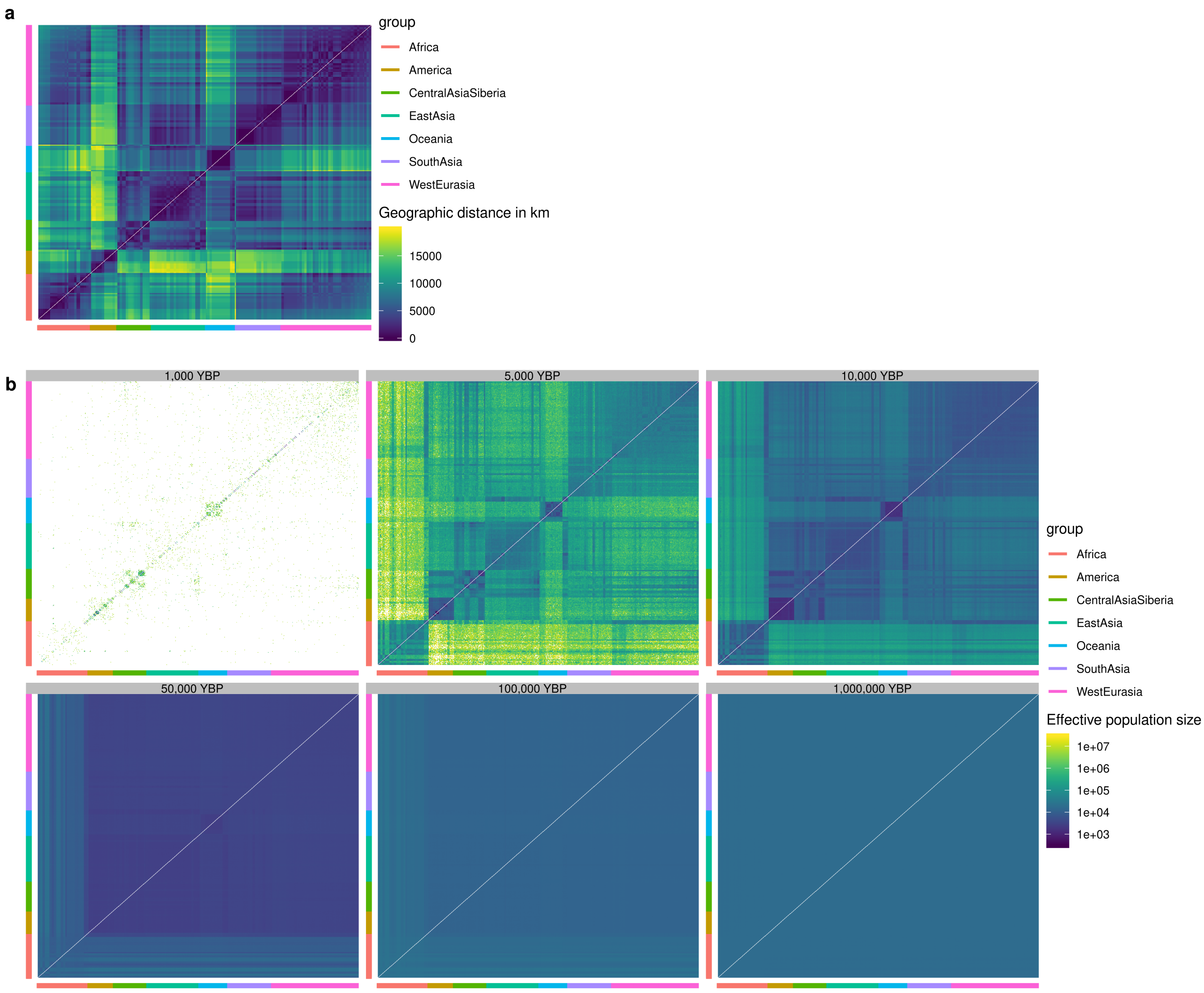

Figure 3: **a**, Geographic distance of haploid sequences in the Simons Genome Diversity Project, with rows and columns sorted by region (as defined in Fig. 1), and then by population. **b**, Diploid effective population size between pairs of haploid sequences in the same order as in **a**, where each matrix corresponds to one epoch.

4 Structure through time

Genealogies contain information on how genetic structure has changed through time. To do this, we first extract coalescence rates for all pairs of haploid sequences from the genealogies. We use the world-wide genealogies, for which precomputed .coal and .bin files can be found in `./precomp_results`. Given these .bin files produced by `RelateCoalescentRate --mode EstimatePopulationSize` as input, we can quickly obtain coalescence rates between all pairs of haploid sequences as follows:

```
#assuming you are in workshop_materials/24_relate/popsizes/
cd ..
mkdir structure
cd structure

chr=15
#rename bin file
cp ../precomp_results/SGDP_ne_chr15.bin  SGDP_ne_chr${chr}_hap.bin

#instead of specifying a poplabels file, type hap to obtain coalescence rates of pairs of
    haplotypes
${PATH_TO_RELATE}/bin/RelateCoalescentRate \
     --mode FinalizePopulationSize \
     -o "SGDP_ne_chr${chr}_hap" \
     --poplabels hap
```

The output file, names `SGDP_ne_chr15_hap.coal` will contain $\binom{556}{2} = 154,290$ lines and can be loaded into R using the `read.coal` function in the relater package.

Exercise 4

(i) Subset the coal file by epoch and plot a heatmap of the pairwise effective population sizes. The output should look similar to Fig. 3. To compare with geographic distance between samples, you may find the `SGDP_structure.RData` data frame in `precomp_results/` useful, which can be loaded into R using `load("SGDP_structure.RData")`. You can find an example R script for plotting this in `Rscripts/plot_SGDP_structure.R`.

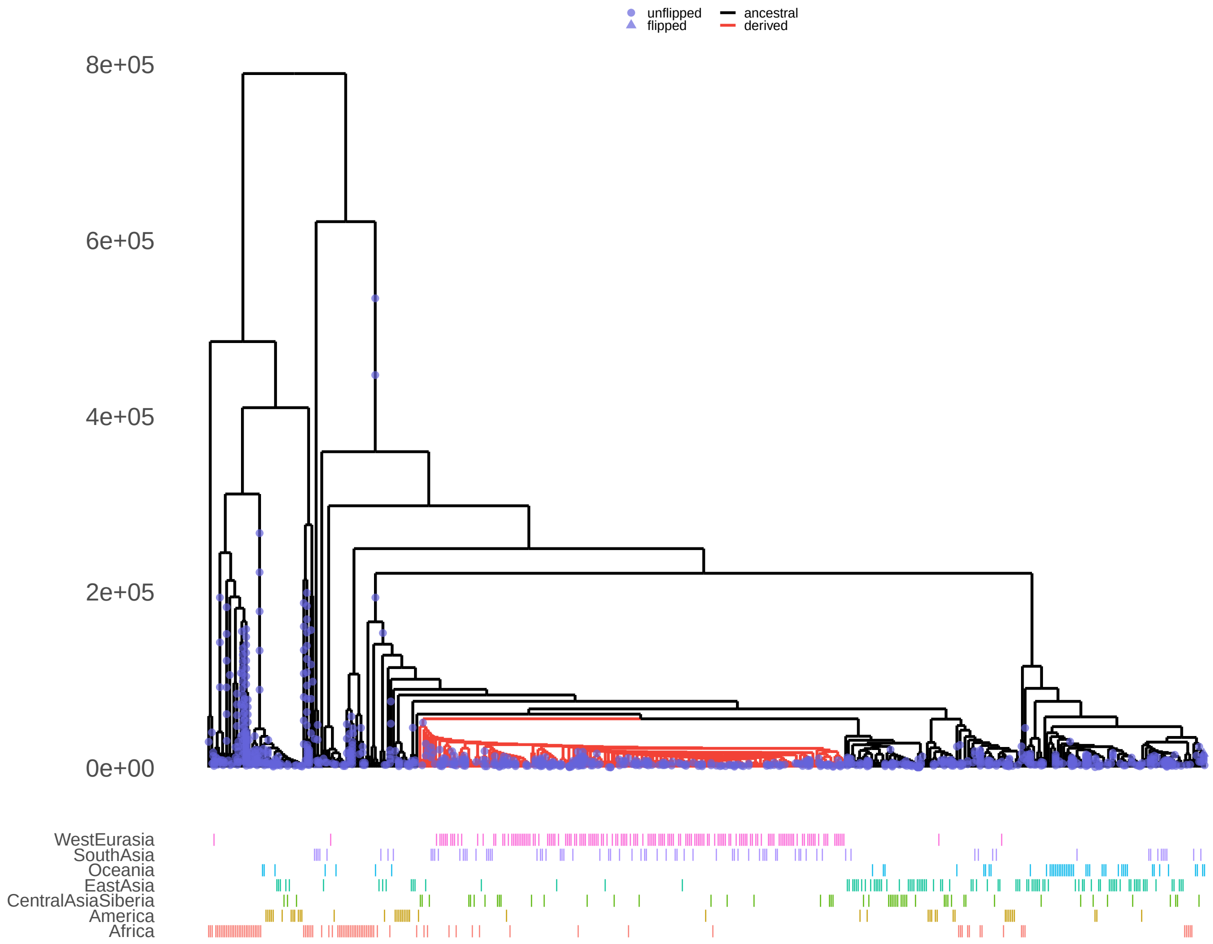

Figure 4: Marginal genealogical tree for rs1426654, a non-synonymous substitution associated with light skin pigmentation. The SNP is located in the SLC24A5 gene on chromosome 15 (BP=48,426,484). Derived allele carriers are shown in red.

5 Detecting evidence for positive selection

Positive natural selection on a derived allele is expected to lead to this allele spreading rapidly in a population, reflected in a burst of coalescence events. A well-known example is positive selection of a variant associated with lighter skin pigmentation in Europe and South Asia (Fig. 4). We have implemented a simple statistic that captures such events and measures the extend to which a mutation has out-competed other lineages.

To analyse selection, we will first apply functions provided with Relate to extract the relevant statistics, and then use the R package relater to analyse the output.

Note 9

We subset the sample into subgroups (e.g., WestEurasia) because the selection p-values assume a panmictic population and may get confounded by genetic structure.

In principle, the selection p-values are robust to misspecification of demographic histories, so we can use genealogies assuming a constant population size here, although we will use genealogies with jointly fitted population sizes and branch lengths below.

```
#assuming you are in workshop_materials/24_relate/structure
cd ..
mkdir selection
cd selection

chr=15
pop="WestEurasia"
output="SGDP_${pop}_ne_chr${chr}"

# Extract genealogies corresponding to pop (here WestEurasia)
${PATH_TO_RELATE}/bin/RelateExtract \
     --mode SubTreesForSubpopulation \
     --pop_of_interest ${pop} \
     --poplabels "../data/SGDP_region.poplabels" \
     --anc "../precomp_results/SGDP_all/SGDP_all_ne_chr${chr}.anc.gz" \
     --mut "../precomp_results/SGDP_all/SGDP_all_ne_chr${chr}.mut.gz" \
     -o ${output}

# Using these genealogies, calculate frequencies through time
# This will output a *.freq and *.lin file
${PATH_TO_RELATE}/bin/RelateSelection \
     --mode Frequency \
     -i ${output} \
     -o ${output}

# Next, calculate selection p-values.
# This will take the output of the previous step as input and output a *.sele file.
${PATH_TO_RELATE}/bin/RelateSelection \
     --mode Selection \
     -i ${output} \
     -o ${output}

# Also calculate a *.qual file storing statistics about the quality of trees
# (e.g., number of mutations mapping to the tree)
${PATH_TO_RELATE}/bin/RelateSelection \
     --mode Quality \
     -i ${output} \
     -o ${output}
```

Once we have generated these files, we use relater to analyse them in R.

```r
output <- "./SGDP_WestEurasia_ne_chr15"

# parse files
mut  <- read.mut(paste0(output, ".mut"))
freq <- read.freq(paste0(output, ".freq"))
sele <- read.sele(paste0(output, ".sele"))
qual <- read.qual(paste0(output, ".qual"))

# combine these into a single data frame called allele_ages
allele_ages <- get.allele_ages(mut = mut, freq = freq, sele = sele)
# use the *qual file to filter out SNPs mapping to "bad" trees
allele_ages <- filter.allele_ages(allele_ages, qual)

head(allele_ages)
```

Exercise 5

Analyse the `allele_ages` data frame. What is the SNP with the lowest selection p-value? How do the selection p-values differ for trees assuming a constant population size?