MOSAIC

Michael Salter-Townshend

2021-07-01

MOSAIC Organises Segments of Ancestry In Chromosomes

Reference

Fine-Scale Inference of Ancestry Segments Without Prior Knowledge of Admixing Groups Salter-Townshend, M. and Myers, S.R. **Genetics** (2019): 10.1534/genetics.119.302139

For a result browser on 95 human populations see https://maths.ucd.ie/~mst/MOSAIC/HGDP_browser/.

Overview

MOSAIC is a tool for modelling multiway admixture using dense genotype data. Given a set of potentially admixed haplotypes (targets) and multiple labelled sets of potentially related haplotypes (panels), MOSAIC will infer the most recent admixture events occurring in the targets in terms of the panels.

It is not necessary that any of the panels are good direct surrogates for the unseen mixing populations as MOSAIC will infer parameters controlling:

- Recombination rates before and after admixture (ρ and Π respectively).
- The stochastic relationship between panels and ancestral populations (μ).
- Timings (from coancestry curves) and ancestry proportions (α) of the admixture events.
- Mutation / error rates for the haplotypes (θ).

Phasing improvements in light of the admixture model are performed and **local ancestry along the genome** is estimated.

Inputs / Data

The `example_data` folder packaged with MOSAIC contains example human data for chromosomes 18 to 22.

Examination of the files in the `example_data` folder should make the format of each of the below clear.

As inputs MOSAIC requires:

- Phased haplotypes for reference panels and target (admixed) individuals (MOSAIC attempts to detect and correct phasing errors in targets): these should be named `pop.genofile.chr` where pop is the population (panel) name and chr is the chromosome index.
  All entries should be $0, 1, ?$ indicating ref, alt, or missing entries respectively. The rows are #snps and the columns are #haps. **Note that there should be no spaces in these files.**
- A population names file: `sample.names` format unimportant apart from first column should have all the population names.
- SNP files: `snpfile.chr` with #snps rows and 6 columns comprising rsID, chr, distance, position, allele ?, allele ?.

- Recombination rates files: `rates.chr` 3 rows of #sites, position, cumulative recombination rate (in centiMorgans).

Outputs / Results

- A folder called `MOSAIC_RESULTS` is required to hold log-files (foo.out) and results (foo.RData).
- A folder called `MOSAIC_PLOTS` is required to hold the plots created by default by a MOSAIC run.
- A folder called `FREQS` is required to hold the frequencies used to compute F_{st} statistics if required.

The above will be created on Linux systems as required if not present.

Parameters Inferred

There are 4 sets of parameters inferred via EM:

1. Π: prob. of switching between latent ancestries, including switch to same anc; dimension $A \times A$ where A is the number of mixing groups.
2. ρ: prob. of switching haps within each ancestry; scalar.
3. μ: copying matrix; μ_{ia} is the probability of a donor from group i given ancestry a; dimension $K \times A$ where K is #donorpops.
4. θ: prob. of a difference b/w copied and copying haps at a locus; scalar.

See reference for details.

Simple Simulation Study

The following demonstrates MOSAIC via a quick and simple simulation that takes about 5 minutes to run, involving 2-way admixture between English and Mandenkan genomes 30 generations ago in approximately equal proportions on chromosomes 18 to 22. When the seed is set in *R* as below, identical results and plots should be obtained.

```
require(MOSAIC)
#> Loading required package: MOSAIC
set.seed(123)
```

To run the simulation and output default results and plots, run either:

```
Rscript mosaic.R simulated example_data/ -c 18:22 -n 3 -p "English Mandenka" -gens 30
```

or equivalently in an interactive *R* session:

```
mosaic.result=run_mosaic("simulated","../example_data/",chrnos=18:22,A=2,NUMI=3,
                         pops=c("English","Mandenka"),gens=30)
#> using 2 cores
#> Admixing 3 individuals from English and Mandenka genomes 30 generations ago
#> creating admixed Chr 18
#> mapping chr 18 to a grid...
#> Finding number at each location on chr 18 ...
#> Mapping true ancestry array for chr 18 to the grid
#> creating admixed Chr 19
#> mapping chr 19 to a grid...
#> Finding number at each location on chr 19 ...
#> Mapping true ancestry array for chr 19 to the grid
#> creating admixed Chr 20
#> mapping chr 20 to a grid...
#> Finding number at each location on chr 20 ...
#> Mapping true ancestry array for chr 20 to the grid
#> creating admixed Chr 21
#> mapping chr 21 to a grid...
#> Finding number at each location on chr 21 ...
#> Mapping true ancestry array for chr 21 to the grid
#> creating admixed Chr 22
#> mapping chr 22 to a grid...
#> Finding number at each location on chr 22 ...
#> Mapping true ancestry array for chr 22 to the grid
#>
#> Fitting model to 3 simulated 2-way admixed target individuals using 3 panels
#> EM inference is  on  and re-phasing is  on
#> Initialise parameters of MOSAIC based on ancestry unaware copying probabilities
#>
#> Fitting no-ancestry model
#> 2 %:  -70739.72 ( NaN )
#> 4 %:  -70330.05 ( 409.6689 )
#> 6 %:  -70112.55 ( 217.5004 )
#> 8 %:  -70003.39 ( 109.161 )
#> 10 %:  -69951.01 ( 52.37735 )
#> 12 %:  -69926.59 ( 24.41881 )
#> 14 %:  -69915.35 ( 11.24038 )
#> 16 %:  -69910.17 ( 5.187935 )
#> 18 %:  -69907.73 ( 2.437576 )
#> 20 %:  -69906.54 ( 1.183465 )
#> 22 %:  -69905.94 ( 0.6016791 )
#> 24 %:  -69905.62 ( 0.3232508 )
#> 26 %:  -69905.44 ( 0.1839838 )
#> 28 %:  -69905.32 ( 0.1104629 )
#> 30 %:  -69905.26 ( 0.06933981 )
#> 32 %:  -69905.21 ( 0.04504064 )
#> 34 %:  -69905.18 ( 0.02999608 )
#> 36 %:  -69905.16 ( 0.02033609 )
#> 38 %:  -69905.15 ( 0.01396591 )
#> 40 %:  -69905.14 ( 0.009685168 )
#> EM iterations have converged
#>
#> Initialising copying matrix Mu; 30 gridpoints per 0.5 cM width window
#> Fitting mixture model of switch counts in windows
#> EM converged in mixture model for initialising copying matrix Mu
#> thinning to at most 100 donors at each gridpoint
#> ######################## round  1 of  5 ######################
#> 10 %:  -69966.13 ( NaN )
#> 20 %:  -69610.56 ( 355.5737 )
#> 30 %:  -69428.91 ( 181.6457 )
#> 40 %:  -69331.17 ( 97.74584 )
#> 50 %:  -69276.45 ( 54.71852 )
#> 60 %:  -69244.63 ( 31.81959 )
#> 70 %:  -69225.44 ( 19.1825 )
#> 80 %:  -69213.51 ( 11.93255 )
#> 90 %:  -69205.9 ( 7.616496 )
#> 100 %:  -69200.93 ( 4.962176 )
#> thinning to at most 100 donors at each gridpoint: log-likelihood -69200.93 -> -69188.04
#> re-phasing... 1264  phase flips made after an average of  4.133333 hunts/ind/chromosome: log-likelih
#> ######################## round  2 of  5 #######################
#> 10 %:  -67510.36 ( 40.00323 )
#> 20 %:  -67500.91 ( 9.449373 )
#> 30 %:  -67497.68 ( 3.231248 )
#> 40 %:  -67496.26 ( 1.416351 )
#> 50 %:  -67495.54 ( 0.7275688 )
#> 60 %:  -67495.12 ( 0.4104172 )
#> 70 %:  -67494.88 ( 0.2437752 )
#> 80 %:  -67494.73 ( 0.1490075 )
#> 90 %:  -67494.64 ( 0.09268921 )
#> 100 %:  -67494.58 ( 0.0583771 )
#> thinning to at most 100 donors at each gridpoint: log-likelihood -67494.58 -> -67464.01
#> re-phasing... 36  phase flips made after an average of  1.733333 hunts/ind/chromosome: log-likelihoo
#> ######################## round  3 of  5 #######################
#> 10 %:  -67450.08 ( 1.450651 )
#> 20 %:  -67449.45 ( 0.6300403 )
#> 30 %:  -67449.14 ( 0.3165053 )
#> 40 %:  -67448.96 ( 0.1749287 )
#> 50 %:  -67448.86 ( 0.1028006 )
#> 60 %:  -67448.8 ( 0.0629838 )
#> 70 %:  -67448.76 ( 0.03981518 )
#> 80 %:  -67448.73 ( 0.02582572 )
#> 90 %:  -67448.71 ( 0.01713126 )
#> 100 %:  -67448.7 ( 0.01159279 )
#> thinning to at most 100 donors at each gridpoint: log-likelihood -67448.7 -> -67446.19
#> re-phasing... 17  phase flips made after an average of  1.066667 hunts/ind/chromosome: log-likelihoo
#> ######################## round  4 of  5 #######################
#> 10 %:  -67445.05 ( 0.3809628 )
#> 20 %:  -67444.85 ( 0.2073859 )
#> 30 %:  -67444.73 ( 0.1203078 )
#> 40 %:  -67444.65 ( 0.07311457 )
#> 50 %:  -67444.61 ( 0.0459981 )
#> 60 %:  -67444.58 ( 0.02974981 )
#> 70 %:  -67444.56 ( 0.01969171 )
#> 80 %:  -67444.54 ( 0.01329617 )
#> 90 %:  -67444.54 ( 0.00913493 )
#> EM iterations have converged
#> thinning to at most 100 donors at each gridpoint: log-likelihood -67444.54 -> -67442.12
#> re-phasing... 22  phase flips made after an average of  1.2 hunts/ind/chromosome: log-likelihood -67
#> ######################## round  5 of  5 #######################
#> 10 %:  -67435.14 ( 0.1200687 )
#> 20 %:  -67435.07 ( 0.06975848 )
#> 30 %:  -67435.03 ( 0.04257603 )
#> 40 %:  -67435 ( 0.02704066 )
#> 50 %:  -67434.98 ( 0.01768534 )
#> 60 %:  -67434.97 ( 0.01183777 )
#> 70 %:  -67434.96 ( 0.00807596 )
#> EM iterations have converged
#> thinning to at most 100 donors at each gridpoint: log-likelihood -67434.96 -> -67433.93
#> re-phasing... 20  phase flips made after an average of  1.133333 hunts/ind/chromosome: log-likelihoo
#> run one final round of EM
#> 0 %:  -67431.57 ( 0.06285094 )
#> 1 %:  -67431.54 ( 0.03622428 )
#> 2 %:  -67431.52 ( 0.02302414 )
#> 2 %:  -67431.5 ( 0.01527851 )
#> 2 %:  -67431.49 ( 0.01037149 )
#> 3 %:  -67431.48 ( 0.007159926 )
#> EM iterations have converged
#> saving localanc results to file
#> calculating ancestry aware re-phased coancestry curves
#> calculating Fst values
#> Saving frequencies of simulated data
#> Saving SNP frequencies of French
#> Saving SNP frequencies of Moroccan
#> Saving SNP frequencies of Yoruba
#> saving final results to file
#> Expected r-squared (genomewide): 0.9291456
#> Actual r-squared (genomewide): 0.8806214
#> Fst between mixing groups:
#> anc1xanc2
#> 0.1595527
#> Rst between mixing groups:
#> anc1xanc2
#> 0.2470911
#> saving plots to MOSAIC_PLOTS/ folder
```

Once this has completed, MOSAIC will have done:

- Simulated 2-way admixture using English and Mandenkan chromosomes 30 generations ago.
- Read in these simulated chromosomes, along with all available reference panels in the `example_data` folder.
- Inferred the model parameters (μ, θ, Π, ρ) via EM.
- Corrected phasing errors that scramble local ancestry.
- Estimated 2-way local ancestry along each chromosome for each admixed individual.
- Estimated F_{st} between each ancestral group and each ancestral group and each reference panel and R_{st} (see paper).
- Saved key plots to `MOSAIC_PLOTS` as PDFs.
- Saved all results to `MOSAIC_RESULTS`.

Loading Results

The results can be loaded in an R session using:

```
load("MOSAIC_RESULTS/simulated_2way_1-3_18-22_148_60_0.99_100.RData") # model parameters, etc
load("MOSAIC_RESULTS/localanc_simulated_2way_1-3_18-22_148_60_0.99_100.RData") # local ancestry
```

If MOSAIC has been run within R using the `run_mosaic()` command above then you can alternatively use

```
attach(mosaic.result)
```

to attach the results for further use while in the same session (or after quitting R and saving the workspace).

Plots

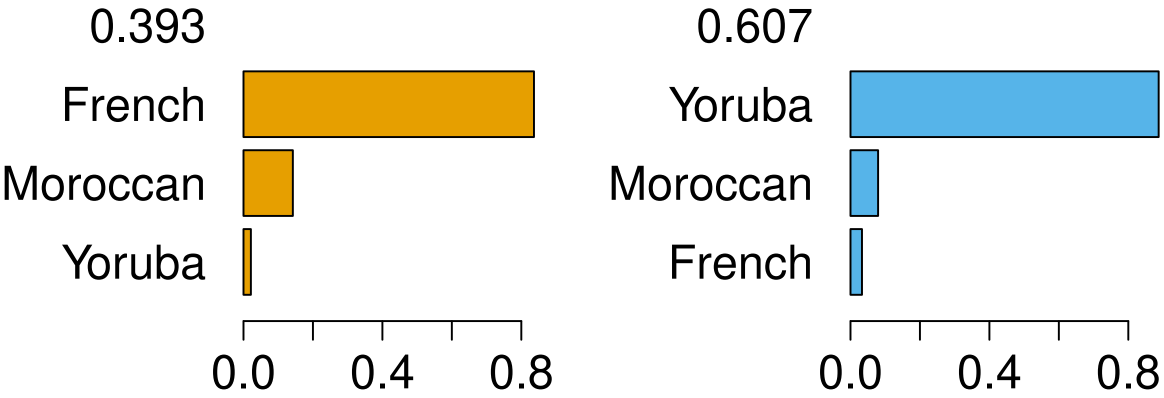
After loading (or attaching) the results in *R*,
each of the plots can be created within *R* by running:

```
plot_all_mosaic(pathout="MOSAIC_PLOTS/",target)
```

to output default plots to the folder `MOSAIC_PLOTS/`. Note that this is already run automatically by default within `run_mosaic()`

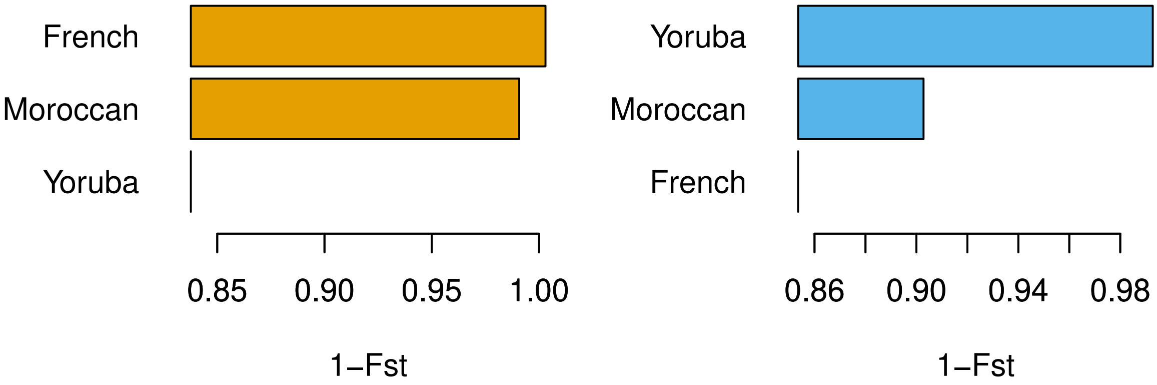
Or you can generate each plot individually:

```
ord.Mu=plot_Mu(Mu,alpha,NL)
```



Inferred Copying Matrix μ. One ancestry (that generated from English segments) mostly copies haplotypes in the French panel and the other (Mandenkan) mostly copies haplotypes from the Yoruban panel, as expected.

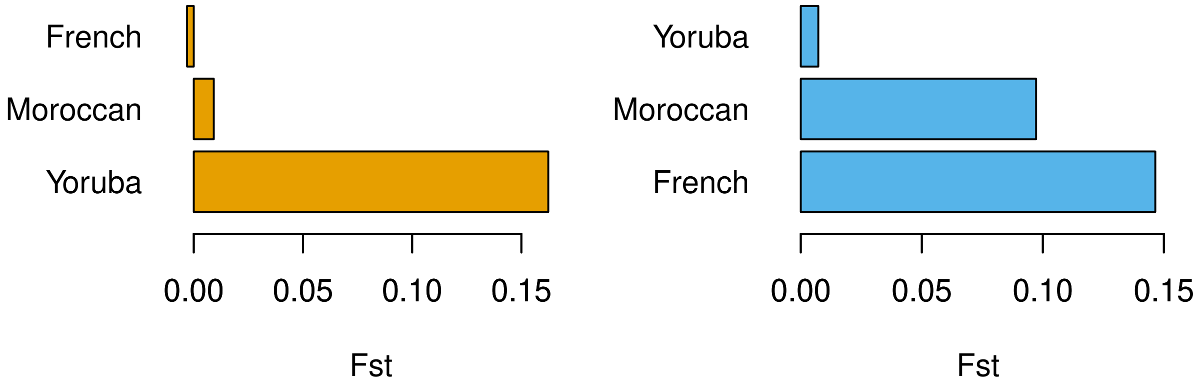
```
ord.Fst=plot_Fst(all_Fst$panels,ord=T)
```



$1 - F_{st}$ estimates between each ancestral group and each donor panel; closer population pairs are larger.

This can be flipped to show the actual F_{st} values by including the `reverse=FALSE` argument:

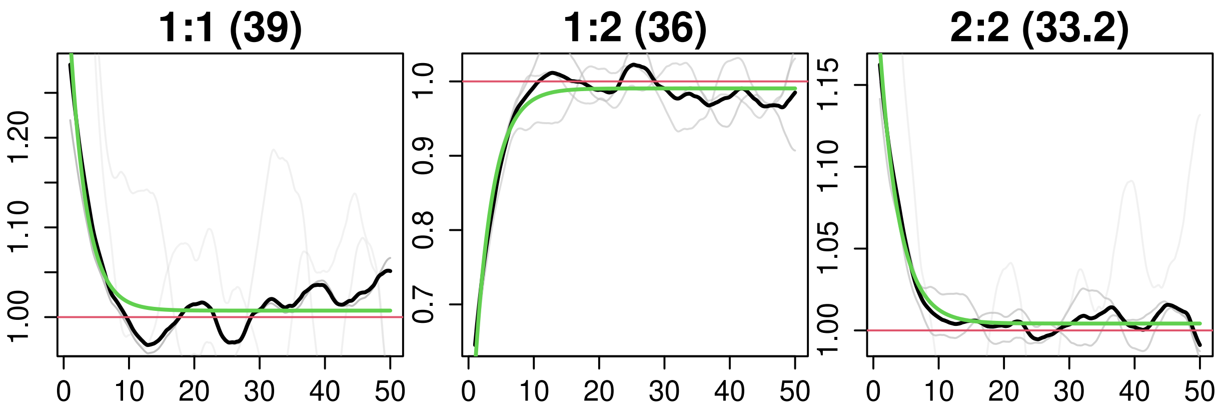
```
tmp=plot_Fst(all_Fst$panels,reverse=FALSE)
```



Raw unordered F_{st} values; closer population pairs are smaller. In this case we have also not reordered the panels in order of closeness under F_{st}.

Plot co-ancestry curves used to infer event timings:

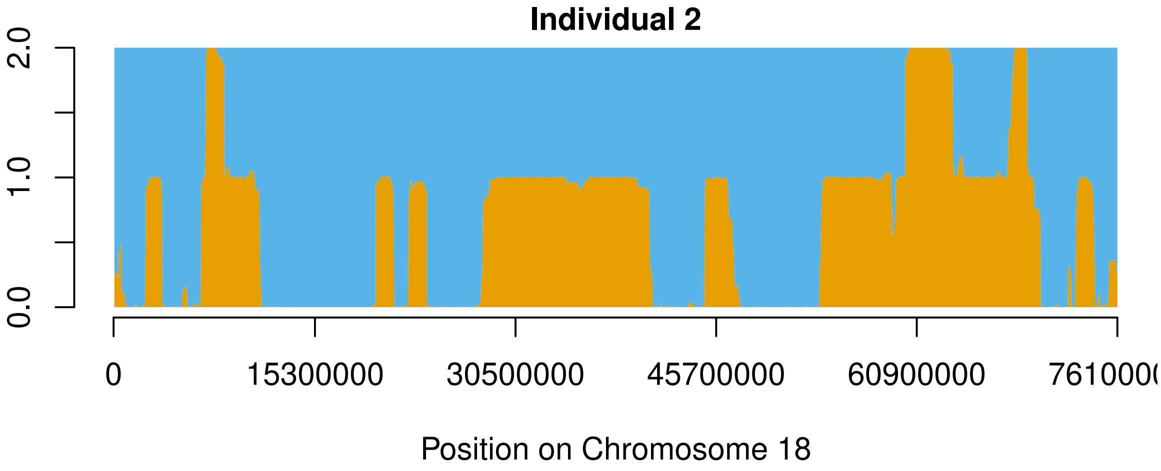
```
fitcc=plot_coanccurves(acoancs,dr)
```



Inferred Coancestry Curves. `dr` is the gap in genetic distance between successive gridpoints. These coancestry curves are somewhat rough at longer distances as we have only used short chromosomes in the analysis. The black lines are empirical coancestry curves across all target individuals, the light grey are per individual, and the green is the fitted single-event coancestry curve.

Look at the 2^{nd} individuals first chromosome:

```
chr=1
ind=2
dipplot(chr,ind,g.loc[[chr]],ind,localanc,xlab=paste("Position on Chromosome",chrnos[chr]),ylab="")
mp<-axTicks(1,round(axp=c(min(g.loc[[chr]]),max(g.loc[[chr]]),5)))
axis(1,at=mp,labels=signif(mp,3))
```



Local ancestry estimates. The second and third line add details to the axes, etc.

You can cycle quickly through all individuals and all chromosomes using:

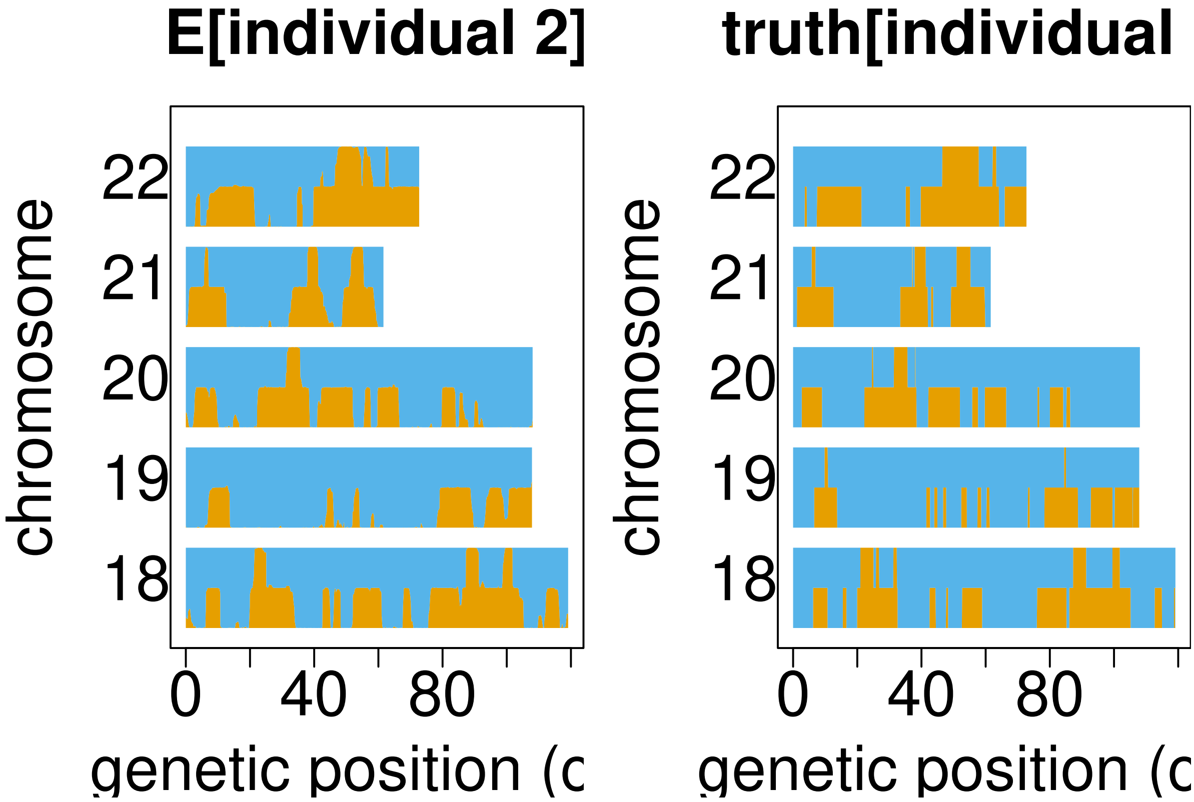
```
plot_localanc(chrnos,g.loc,localanc,g.true_anc)
```

and hit return to display each new plot.

When any such second set of local ancestries is provided, both are plotted and the Pearson correlation r^2 between them over chromosomal positions is reported for each individual on each chromosome. Here we have supplied the true local ancestry `g.true_anc`, known as this is simulated data.

Alternatively, create a karyogram (local ancestry of entire genome) for a given individual using:

```
ind=2
karyogram(A, chrnos, localanc, g.loc, GpcM, ind, dist = "genetic", g.true_anc = g.true_anc)
```



The karyogram can also be plotted using physical distances by setting dist=“physical”.

Accuracy of estimated local ancestry

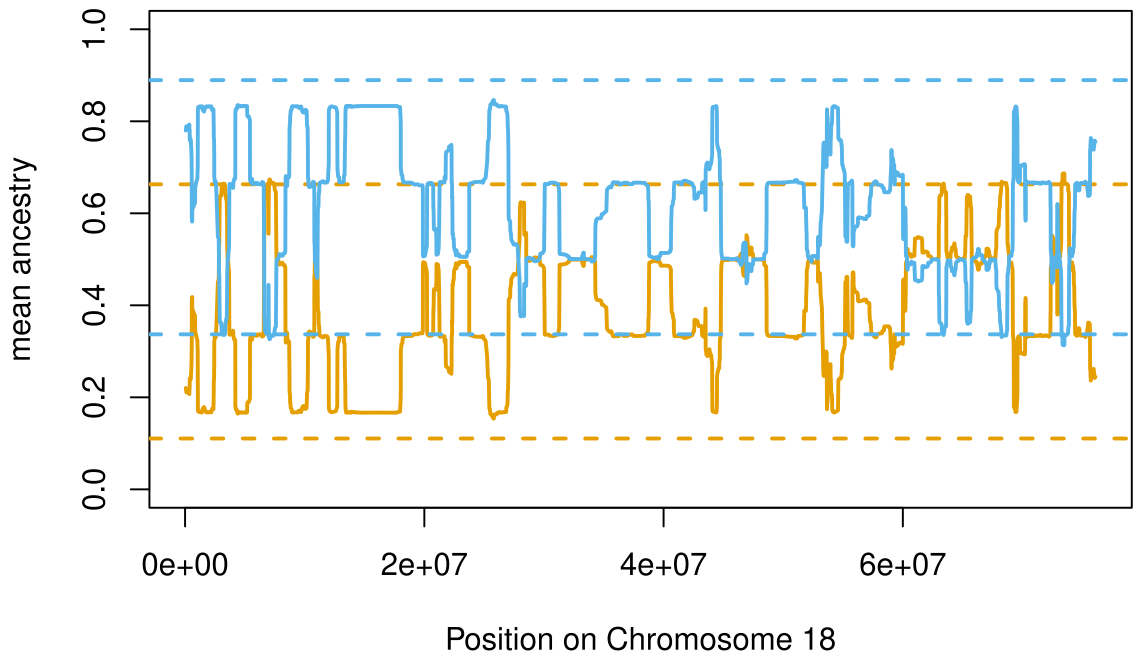
The object `localanc` is a list; each entry represents one chromosome, each of which is an array of dimension $A \times 2N \times G$ where A is the number of modelled ancestries (A-way admixture), N is the number of target individuals, and G is the number of gridpoints (evenly spaced positions on genetic distance) on that chromosome. The a, h, g entry of `localanc[[chr]]` is therefore the inferred probability that haplotype $(h+1)\%\%2+1$ of individual $(h+h\%\%2)/2$ is of ancestry a at the g^{th} point of chromosome `chr`.

The expected number of alleles on each diploid individual can be quickly found using

```
dip_localanc=dip(localanc)
```

Plots of the mean (across individuals) local ancestry can be made using

```
chr=1
m_localanc=plot_mean_localanc(chr,chrnos,g.loc,localanc,ret=TRUE)
```



which will plot the mean for each ancestry with a different colour on chromosome `chr`. Here the non-default setting `ret=TRUE` has been used to also return mean values in the form of a list. Each element represents the mean local ancestry for an ancestry across all gridpoints on that chromosome.

Expected accuracy of local ancestry $\mathbf{E}[r^2]$ is calculated using

```
dip_expected_fr2(localanc)
#> [1] 0.9291456
```

and accuracy of local ancestry estimation in the presence of a known truth given by `g.true_anc` is provided by

```
dip_fr2(localanc,g.true_anc)
#> [1] 0.8806214
```

which calculates r^2 across all target individuals and all chromosomes analysed.

Local ancestry at SNP positions

The above calculations and plots show local ancestry along evenly spaced gridpoints on recombination distances. You can get local ancestry estimates at the SNP positions using:

```
local_pos=grid_to_pos(localanc,"../example_data/",g.loc,chrnos)
```

where the SNP positions you'd like to map back are read from the relevant `snpfiles` to and this is for the first chromosome for which local ancestry has been estimated. Naturally, local ancestry estimation accuracy will be slightly different at the SNP positions (typically higher as these are where we have genotype information)

```
true_anc_pos=grid_to_pos(g.true_anc,"../example_data/",g.loc,chrnos)
dip_fr2(local_pos,true_anc_pos)
#> [1] 0.8909004
```

Other Options

- MOSAIC will use any additional groups found in the data folder as donor panels but these can also be specified manually as follows:

```
mosaic.result=run_mosaic("simulated","../example_data/",18:22,A=2,NUMI=3,
             pops=c("English","Mandenka", "French", "Yoruba"))
```

so that only `French` and `Yoruba` are used here. When the first argument (the target) is `simulated` then the first A populations are used to simulate admixed chromosomes (and haplotypes not used for the simulated targets aren't used for inference) but when the target is not `simulated` then all specified `pops` are used as reference panels.

- To create a version of the copying matrix μ plot that uses bar colour densities rather than bar lengths to represent copying proportions, use:

```
ord.Mu<-plot_Mu(Mu,alpha,NL,showgradient=TRUE)
```

- For help use

```
Rscript mosaic.R --help
```

on the command line to list all arguments to MOSAIC or

```
?run_mosaic
```

within R for help on the main function.

email michael.salter-townshend@ucd.ie for help