Ancestry HMM Manual

Version 0.94

Russ Corbett-Detig

Download and Compile:

```
$ git clone https://github.com/russcd/Ancestry_HMM.git
$ cd Ancestry_HMM/src/
$ make
```

Dependencies:

As of version >0.9, the software requires the C++ linear algebra library, armadillo. More information and detailed download instructions can be found here, http://arma.sourceforge.net/. Using homebrew on OSX, this software can be installed using

```
$ brew install homebrew/science/armadillo
```

If you cannot get a root installation of armadillo, the following stackoverflow link is helpful for linking a locally installed armadillo during compilation: http://stackoverflow.com/questions/10168181/armadillo-installation.

It is also recommend that users install the google-perftools package as compilation using tcmalloc tends to decrease runtimes, sometimes substantially. However, this is not necessary to use the software.

Disclaimer:

The current version includes functionality for fitting multiple pulse ancestry models. We are currently thoroughly testing this option and it is considered experimental. Single pulse models perform as described previously in Corbett-Detig and Nielsen (2017).

Basic Usage:

```
$ ancestry_hmm (options) –i [input_file] –s [sample_file]
```

Detailed Options and Usage:

Required: Overall ancestry proportion

```
-a [int] [double] [double] …
```

This option specifies the number of ancestral populations (the first argument) and the overall ancestry proportion of each in the sample. Note that the number of ancestry proportions specified must exactly equal the number given following –a. For example, if there are three ancestry types in a given population, the first at proportion 0.6 and the others at proportion 0.2, the following command line argument would be used:

```
-a 3 0.6 0.2 0.2
```

In general, it is straightforward to estimate the global ancestry proportions in an admixed population using available software packages such as Admixture, NGSAdmix, or Structure, we therefore recommend using one of these software packages prior to working on local ancestry inferences using this software (however, see section XXX below).

Required: Ancestry Pulses

```
-p [int, ancestry type] [int, time before present] [double, proportion of ancestry]
```

Minimally, two ancestry pulses should be specified when running this software, each using the syntax above. The first argument specifies the ancestry type associated with the pulse, where 0 indicates the first ancestry population in the input file, 1 indicates the second, 2 the third, etc. The second parameter is the time before the present (in generations) at which the ancestry pulse occurred. If time is provided as a positive number, it will be treated as fixed. When time is provided as a negative number, it is treated as the initial estimate to be optimized when fitting the admixture model. Finally, the third parameter specifies the proportion of the present ancestry of the sample that resulted from this pulse. Here again, if this is provided as a negative number, the parameter will be estimated if applicable.

For example, to fit a model with a single pulse of ancestry type 1 into a population of ancestry type 0, the following could be used. Here both ancestry proportions are 0.5.

```
-p 0 100000 0.5 -p 1 -100 0.5
```

The program would then estimate the time that the ancestry pulse (which replaced 50% of individuals) using an initial time estimate of 100 generations. If each ancestry type enters the population through a single pulse, there is no proportion to be estimated (even if provided as a negative value) and the program will assume that the pulse included all ancestry of that type.

In general, the initial ancestry type of the population, prior to ancestry pulses should be specified as an ancestry pulse with a time of the pulse set to be greater than the maximum time allowable in the program (--tmax, below).

A more complicated model might include three ancestry pulses, where a population initially composed of ancestry type 0 experiences three subsequent pulses from ancestry type 1, 2 and then 1 again. If the time of all pulses, and relative proportions of ancestry in pulses one and three are to be estimated, this would be specified as

```
-p 0 1000000 0.4 –p 1 -1000 -0.2 –p 2 -500 0.2 –p 1 -250 -0.2
```

where this command line would also include the global ancestry proportions

```
-a 3 0.4 0.4 0.2
```

Note that if all of a given ancestral type enters the population through a single pulse, the proportion contributed by that pulse cannot be estimated since by definition it must be equal to the amount of that ancestry type present in the sample.

Required: Input File

Input files are specified as follows with a file format as described in the section that follows.

```
-i <input_file>
```

Required: Sample File

A single file, specifying sample ID's and their ploidy, or path to a ploidy map, must also be provided to the program

```
-s <sample_file>
```

Optional:

-v	Viterbi decoding, default decoding is forward-backward
-g	sample counts provided are genotypes rather than read counts
-b [int] [int]	If bootstrap replicates are to be performed, specify –b and the number of bootstraps and the block size of bootstraps. *E.g.*, "-b 10 1000" would indicate 10 bootstrap replicates each using a block size of 1000 SNPs.
--output-ancestry	If more than one pulse originated in the same source population, these will be summed and the output will be only ancestry states ordered as in the input file. Note that in the current version, this is not compatible with Viterbi decoding, –v, above.
--precision	Set output precision
-r	Number of restarts during nelder-mead search
--tmax	Maximum time of a pulse
--tmin	Minimum time of a pulse
--pmax	Maximum portion of ancestry from one ancestry type, note that this argument and the one below is only relevant for models with two pulses from the same source population.
--pmin	Minimum portion of ancestry from one ancestry type

--tolerance	Difference in log likelihood between worst and best vertices for search termination.
-e	Error rate per site (either genotype or read)
-E	If present on the command line, site specific error rates are provided in the input file. Error rates must appear in the two columns following the recombination rates.
--ne	The effective population size, n. By default this number is multiplied by 2 to accommodate diploid populations. However, for use with autotetraploids for example, it might be reasonable to supply 2n instead.
--fix	Ancestral allele frequencies are fixed, rather than treated as uncertain. This is expected to be useful when parental sequences are known with certainty, for example in performing qtl mapping or in experimental evolution studies.

Sample File Format

As of version 0.9+, all samples must be specified within a single tab-delimited input file. Each line corresponds to a single sample (in the same order as they are present in the input file). For example, in a file with three samples, the first of ploidy 2 and the second of ploidy 1 and the third of ploidy 2 the first two lines would be

```
Sam1   2
Sam2   1
Sam3   2
```

Note that sample IDs must be unique for all runs of the software in a given directory. This is because output files for each will be the sample id followed by ".posterior" (or, ".viterbi" if viterbi decoding is specified). Therefore the two samples above would have their posterior probabilities output to sample1.posterior and sample2.posterior in the working directory from which the software was run.

Optionally, this file may indicate that samples have variable ploidy by including the full path to a file containing the coordinates of various ploidy levels in the sample (see below).

Input File Format

The following lines specify the allele counts in the reference panels and in the samples.

1. Chromosome
2. Position in basepairs
3. Allele counts of allele A in reference panel 0
4. Allele counts of allele a in reference panel 0
5. Allele counts of allele A in reference panel 1

6. Allele counts of allele a in reference panel 1

If there are additional reference panels, they are included following these columns. So, the next panel would be columns 7 and 8, the next 9 and 10, and so on. All of the following column numbers would be augmented by two for each additional reference panel included. The number of reference panels provided must match the number specified on the command line using –a.

7. Distance in Morgans between the previous marker and this position. For the first position on a given chromosome, this may take any value, as it will be ignored.

Following this, an option column is the site-specific error rates for each allele. Here, the first column denotes the error rate where a read (or genotype) that is really an A is reported as a, and vice versa for the following column. If this is provided (via specifying –E on the command line), the following columns numbers should also be augmented by two as well.

Each sample is then represented by two columns with counts corresponding to

8. Read counts of allele A in sample 1
9. Read counts of allele a in sample 1
10. Read counts of allele A in sample 2
11. Read counts of allele a in sample 2

Additional samples could then be represented by additional pairs of columns. The total number of columns following the recombination rate (and optionally, the site specific error rates), should be exactly equal to the number of samples times two.

Output File Format

The output file is also tab delimited. Each sample will have a separate out file, specified via the sample file as described above. The first line of the file defines the state to which each posterior probability corresponds. Specifically, it will begin with "chrom\tposition\t" and then list the states that are represented by columns 3-n below. The format of this file is

1. The chromosome of the observation.
2. The position in basepairs of the observation.
3. The posterior probability of state 0.
4. –n posterior probability for states 1-n.

Alternatively, if '-v' is specified on the command line, this program will perform Viterbi decoding. We do not recommend using this option unless the sample is haploid or very well inbred because tract lengths are not necessarily obtainable otherwise and the full posterior distribution provides more complete information about site-specific ancestry. However, if used, the Viterbi decoding will output a file with the following format

1. The chromosome of the observation
2. The position of the start of the tract
3. The position of the end of the tract
4. The state of the tract

State Format

In the program output, ancestry states are represented by counts of the number of chromosomes in each ancestry pulse (in order that the pulses are entered on the command line). The number of chromosomes counted in each state will sum to the sample ploidy. Therefore, if there are two ancestry types present and the sample is ploidy 2, the following states are possible.

2,0
1,1
0,2

Where the final state would specify that both chromosomes are in the second ancestry type.

If multiple pulses are present from the same ancestry type, this can be output in two ways. Either the pulses can be individually specified, i.e. a model that includes two pulses from the same source population can be specified for a haploid individual as

1,0,0
0,1,0
0,0,1

Where the second line indicates that the segment is from the first pulse, and the third line indicates that the segment is derived from the second pulse.

Alternatively, if the "--output-ancestry" command line option is used, the second and third pulse, being from the same ancestry type, are collapsed into a single ancestry state and the sum of their posterior probabilities would be reported as a single state:

0,1

Variable Sample Ploidy

It is common for chromosomes to be partially inbred in population sequencing applications. In general, it is preferable to model the inbred segments as a single haploid chromosome, and the outbred segments as a diploid chromosome. We have therefore provided a function to given the program a defined ploidy map (a file that specifies which portions of the genome are inbred and outbred along a sample). To do this, the sample file must be modified slightly. Specifically, each line must include the name of the sample, a number less than 0, and the path to a file containing the ploidy map. For example the following:

```
Sam1   -9        Sam1.ploidy
Sam2   -9        Sam2.ploidy
```

Would indicate that two samples, with ploidy maps provide in the currently working directory with the filenames “Sam1.ploidy” and “Sam2.ploidy”.

Each ploidy map file would then have the following format: chromosome, position start, position stop, and ploidy. For example, an individual might have the following ploidy map

```
1         0         35000 1
1         35001 50000 2
1         50001 100000          1
```

If the sample contains exactly one chromosome and is inbred between positions 0 and 35000 and from 50001 to 100000, and outbred between 35001 and 50000.

If a ploidy map is specified the entire genome must be covered by one of the tracts in the file and the chromosomes must be provided in the same order that they appear in the input file.

How to cite this software

Corbett-Detig, R. and Nielsen, R., 2017. A hidden Markov model approach for simultaneously estimating local ancestry and admixture time using next generation sequence data in samples of arbitrary ploidy. *PLoS genetics*, *13*(1), p.e1006529.

```
@article{corbett2017hidden,
title={A hidden Markov model approach for simultaneously estimating local
ancestry and admixture time using next generation sequence data in samples of
arbitrary ploidy},
author={Corbett-Detig, Russell and Nielsen, Rasmus},
journal={PLoS genetics},
volume={13},
number={1},
pages={e1006529},
year={2017},
publisher={Public Library of Science} }
```