MultiWaveInfer v1.0.2

==

Short description:

MultiWaveInfer is designed to scan the number of waves of admixture events, and estimate the parameters of multi-waves, multi-ancestral populations admixture models via the length distribution of the ancestral tracks.

The program works mainly in two steps: Firstly, use EM-algorithm to scan the number of waves for each ancestral population. Secondly, use the theoretical length distribution of ancestral tracks to estimate the parameters (i.e. the proportions and generations).

1.Compilation and installation

1.1 Library dependency

MultiWaveInfer depend on boost library, make sure the boost is installed. For example, boost library can be easily installed in Ubuntu/Debian Linux

```
bash$ sudo apt-get install libboost-dev
```

More details of installation of boost can be found at

```
http://www.boost.org/doc/libs/1_58_0/more/getting_started/unix-variants.html
```

1.2 Compile from source code

It's very easy to compile from the source code by the following commands:

```
bash$ tar -zvxf MultiWaveInfer.tar.gz
bash$ cd MultiWaveInfer/src
bash$ make
```

After compiling, you will get the executable *MultiWaveInfer*, and it's ready to use, just typing the command below to get help information:

```
bash$ ./MultiWaveInfer -h or bash$ ./MultiWaveInfer --help
```

1.3 Installation

Alternatively, you can also copy the executable *MultiWaveInfer* to /usr/local/bin directory:

```
bash$ cp ./MultiWaveInfer /usr/local/bin/
```

2. Test with the toy data

2.1 a simple simulated two waves admixture example

```
bash$ ./MultiWaveInfer --input ../example/two.seg
```

Example explanation:

MultiWaveInfer will read the ancestral tracks from two.seg, after a while, the optimal model and corresponding generation and proportion will print to screen. The format will explained later.

The following is output of the toy data:

```
// COMMAND ./MultiWaveInfer -i ../example/two.seg

Reading data from ../example/two.seg...
Start scan for admixture waves...
Perform scanning for waves of population 2...
Perform scanning for waves of population 1...
Finished scanning for admixture waves.

There is(are) 2 wave(s) of admixture event(s) detected
-------------------------------------------------------------------------------
                         Results summary

          Parental population          Admixture proportion
                   1                   0.506226
                   2                   0.493774


Possible scenario: #1
  24.3692: (0,   0.801602) ==========>||<========== (1,   0.198398) :   24.3692
                                       ||
                                       ||
                                       ||
                                       ||<========== (1,   0.384016) :   11.0706
                                       ||
                                       ||
                                       ||

Hint:
0: population-2; 1: population-1;
-------------------------------------------------------------------------------
```

We use a tree to present the results. The simulated admixed population has two reference populations (population 1 and 2). There are 2 waves of admixture events. The first admixture event was happened in 24 generations ago. The ancestral populations are pop2 and pop1 and corresponding mixture proportions are 0.198398 and 0.801602. The second admixture event was happened in 11 generations ago. The ancestral population and corresponding mixture proportions is pop1 and 0.384016.

User can redirect the output to a file, such as:

```
bash$ ./MultiWaveInfer --input ../example/sim1.seg > sim1_opt.log
```

2.2 A full arguments example

```
bash$ ./MultiWaveInfer -i ../example/three.seg -l 0.01 -a 0.01 -e 0.0001 \
-m 5000 > three_fopt.log
```

Example explanation:

Again, *MultiWaveInfer* read ancestral tracks from file three.seg, discard the tracks shorter than 0.01 Morgan, the significance level of LRT is 0.01,and the convergent condition is 0.0001, and the Max number of iterations to perform EM is 5000. Finally, the outputs will be redirected to three_fopt.log.

3. File format

3.1 Input file format

MultiWaveInfer is easy to use, only need one file, in which each line represents a ancestral track with the start point, end points, from which ancestry the track originates. The start and end points units are in Morgan.

For example:

```
0.00000000    0.34602058    Yoruba
0.34602058    0.34614778    French
......
0.40759031    0.41517938    Yoruba
```

4. Arguments

-i/--input <string>

This argument is required, in which user specify the filename of input ancestral tracks, format described above.

-a/--alpha [double]

This argument is optional, in which user specify the significance level to reject null hypothesis in likelihood ratio test (LRT). Default is 0.001.

-e/--epsilon [double]

This argument is optional, in which user specify epsilon to check whether a parameter converge or not. Default is 0.000001.

-l/--lower [double]

This argument is optional, in which user specify the lower bound to discard short tracks. The default is 0, which does not discard any hort tracks. However, due to method limitation in local ancestry inference, very short tracks are generally not reliable.

-p/--minProp [double]

This argument is optional, in which user specify the minimum survival proportion for a wave at the final generation. Default is 0.05.

-m/--maxIt [integer]

This argument is also optional, in which user specify the maximum number of iterations to scan for waves of admixture events. Default is 10000.

5. Options

-h/--help

Print help message, default is OFF

-s/--simple

Run in simple mode, default is OFF

Here simple mode refer to the scenario in which each parental population contributes only one pulse of admixture (one wave).

6. License

==

7. Questions and suggestions

Questions and suggestions are welcomed, feel free to contact

Shawn xyang619@gmail.com