Tutorial for *AdmixSim* v1.0.3

==========

Short description:

AdmixSim is designed to simulate data for admixed population under various and complex scenarios. With *AdmixSim* user can simulate admixed population with:

1). Arbitrary number of parental populations;

2). Arbitrary wave of population admixture events;

3). Changeable population size generation by generation;

4). Changeable mixture proportion generation by generation.

1.Compile

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

bash$ *tar -zvxf AdmixSim.tar.gz*

bash$ *cd AdmixSim/src*

bash$ *make*

After compiling, you will get the executable *AdmixSim*, just typing the command below to get help information:

bash$ *./AdmixSim -h* or *./AdmixSim --help*

2. Test with the toy data

bash$ *./AdmixSim --file toy.par --length 1.0 --input test --nsample 10 --output sim*1

Example explanation:

AdmixSim will simulate an admixed population which evolves following the model described in file toy.par (details will explained later), with chromosome length 1.0 Morgan. Prefix for parental haplotype file and mapfile is test (i.e. these files are test.hap and test.map). At the end of simulation, 10 haplotypes will be sampled, and the results will be saved to sim1.seg and sim1.hap (file format described later).

3. File formats

3.1 Input file formats

1) Model description file

In model description file, user specify the number of parental haplotypes to copy from; and specify the population size and mixture proportions generation by generation. Below is the format:

Firstly, set up initial number of haplotypes for parental populations, for example:

```
10      20      #two parental populations, first has 10 haplotypes and second has 20
```

Secondly, set up the population size and mixture proportions for each generation in one line, for example:

```
5000    0.9     0.1
5500    0       0.1
......
```

First part and second part are separate by a line with "//". Note: any contents follow "#" are treated as comments

Here is a complete example:

```
#set up number of haplotypes in each parental population
20   30
// #indicate start of second part
1000 0.5 0.5 #initial admixture with two parental population, each contributes 50%
1000 0   0
1000 0.1 0 #second wave admixture, with 10% gene flow from parental population 1
1000 0   0
1000 0   0
1000 0   0.2 #third wave admixture, with 20% gene flow from parental population 2
1000 0   0
1200 0   0 #population size increase to 1200
1000 0   0 #population size decrease to 1000
1000 0   0
......
1000 0   0
```

By the same manner, it's not only quite easy to model one-pulse, two-way admixture as following:

```
20   30
//
1000 0.9 0.1  #generation 1, two parental populations with proportions 90:10
1001 0   0    #generation 2
......
2000 0   0    #generation T
```

---

but also easy to model multiple waves of admixture, or continuous admixture, multiple-way admixture, and so on. It can simulate any model as you want.

Further example 1, multiple-way admixture:

```
10 15   15      20
//
1000    0.2     0.8     0       0       #generation 1
1000    0       0       0       0
......
1200    0       0       0.2     0       #generation t_1, third parental population entered
1200    0       0       0       0
......
1200    0       0       0       0.15    #generation t_2, fourth parental population entered
1200    0       0       0       0
......
1200    0       0       0       0       #generation T, end of simulation
```

Further example 2, discrete multiple-wave admixture:

```
20 15
//
1000    0.2     0.8     #generation 1
1000    0       0
......
1200    0.2     0       #generation t_1, extra wave from parental population 1
1200    0       0
......
1200    0.1     0       #generation t_2, extra wave from parental population 1
1200    0       0
......
1200    0       0.1     #generation t_3, extra wave from parental population 2
1200    0       0
......
1200    0       0.1     #generation T, end of simulation
```

Further example 3, continuous multiple-wave admixture:

```
20 25
//
1200    0.05    0.95    #generation 1
1200    0.05    0        #extra waves from parental population 1 in following generations
......
1200    0.05    0
1200    0.04    0
1200    0.04    0        #generation t_2, proportion changed to 4%
1200    0.04    0
......
1200    0.04    0
1200    0.04    0
......
1200    0.04    0        #generation T, end of simulation
```

Further example 4, another form of continuous multiple-wave admixture:

```
10 15
//
1200    0.15    0.85    #generation 1
1200    0.05    0.01    #extra waves from both in following generations
......
1200    0.05    0.01
......
1200    0.04    0.01    #generation T, end of simulation
```

---

2). Map file

Map file specify the position of the locus, one locus per line, unit in Morgan, here a locus can be a SNP or DNA(RNA) base.

Here is an example:

```
0.00097100
0.00238066
0.00367538
......
```

3). Haplotype file

Haplotype file contains one haplotype per line, the corresponding position for each locus is given in map file, as described above.

Here is an example:

```
10110000001001000010000000101101000101010000110101110000000000010000000011
00100000100001000010000000101101010100110000010111110010000000010000000011
00100000100001000010000000101101010100010100010101110000001000110000000011
00100000100001000010100000101101010100010000010101110000001000100000000011
.....
01000000000001101000000100000001000100000010000000010100010000100001010001
```

In the example, '0' denotes ancestral allele while '1' denotes derived allele, again, the characters can also be a SNP or DNA(RNA) base, for example:

```
AGCTTAGCAGATAGATCGGACGATGATTAGCAGATAGATCGGACGATGATTAGCAGATAGAT
CGGACGATGAC
CTTAGCAGATAGATCGGACGATGATTAGCAGATAGATCGGACGATGATTAGCAGATAGATCG
GACGATGACGA
.....
GCTTAGCAGATAGATCGGACGATGATTAGCAGATAGATCGGACGATGATTAGCAGATAGATC
GGACGATGACT
```

To minimize the number of input files, the haplotypes of parental populations are combined in a single file, which first n1 line corresponds to parental population 1, second n2 line corresponds parental population 2, and so on. Here n1, n2, ... denote the number of parental population haplotypes setting in model description file.

3.2 Output file formats

1). Haplotype file for admixed population

At the end of simulation, N haplotypes are randomly sampled from the admixed population, the format is the same as haplotype file describe above.

2). Ancestral track file

An ancestral track specifies the start point, end point, and from which ancestry the track originates. With these information, user can easily track the ancestry of a segment in chromosome, derive the length of ancestral tracks, the recombination break points, the segment identical by descent (IBD) and so on. With these information, use can perform a lot of statistical analysis and (or) inference.

Notes: only save tracks from the N haplotypes sampled, as described above.

Here is an example:

```
0.00000000      0.07785695      2
0.07785695      0.30178126      1
......
0.30178126      0.41594482      2
```

4. Full argument list

-f/--file model description file [required]

-i/--input prefix of input file [required]

-l/--length length of chromosome simulated [optional, default=1]

-n/--nsample number of individuals sampled [optional, default=10]

-o/--output prefix of output files [optional, default=output]

-s/--seed seed of random generator [optional, default=time]

-h/--help

if you forget the usage of any arguments, don't hesitate to use this one.

-f/--file <filename>

This argument is required, in which to specify the model description file

-i/--input <prefix>

This argument is required, in which user specify the prefix of the input map file and haplotype file, i.e. prefix.map and prefix.hap.

-l/--length [value]

This argument is optional, in which user can specify the length of chromosome to be simulated, unit in Morgan. Default is 1.0 Morgan.

-n/--nsample [value]

This argument is optional, in which user can specify the number of haplotypes to be sampled at the end of simulation.

-o/--output [output_filename]

This argument is optional, in which user can specify the prefix of output files, i.e. output.hap and

output.seg. Default is "output" .

-s/--seed [value]

This argument is also optional, in which user can specify the seed of random generator. The user can use the same random seed to replicates simulation. Default seed is set to current system time.

5. Possible concerns

5.1 About the parental haplotypes

In our simulator, we do not simulate parental haplotypes, the parental haplotypes can be either obtained from other simulator such as ms, or obtained from real dataset. As there exists a lot of simulator can simulate parental haplotypes (sequences), we focus our attention on simulating data for admixed population, in a finer scale.

5.2 About the drift of parental populations

In our simulation, only the haplotypes and tracks from admixed population are sampled and saved. However, our simulator can also simulate the drift of parental population, just modify the model description file as below:

```
50      #number of input haplotypes
//
5000    1       #start generation
5000    0       #second generation
......
5000    0       #generation T, end of simulation
```

---

other procedures are the same as these simulating admixed population.

6. License

GNU GENERAL PUBLIC LICENSE Version 3

http://www.gnu.org/licenses/gpl-3.0.html

==========

7. Questions and suggestions

Questions and suggestions are welcomed, feel free to contact

Shawn xyang619@gmail.com