Manual for AdmixSim 2

1. Introduction

AdmixSim 2 is an individual-based forward-time simulation tool that can flexibly and efficiently simulate population genomics data under complex evolutionary scenarios. It is based on the extended Wright-Fisher model, and it implements many common evolutionary parameters to involve gene flow, natural selection, recombination, and mutation. *AdmixSim 2* can be used to simulate data of dioecious or monoecious populations, autosomes, or sex chromosomes. We have developed Cpp and Python version for *AdmixSim 2*. In addition, we provide an script (vcf2hap.py) to convert the $.vcf$ format to input data format, an script (hap2vcf.py) to convert the output data format to the $.vcf$ format, and an script (Summary_Statistics.py) to calculte common summary statistics: $Tajima'sD$, θ_K, θ_π, and $haplotype\ diversity$. We recommend the Cpp version which is much faster.

The *AdmixSim 2* software is available at https://www.picb.ac.cn/PGG/ or https://github.com/Shuhua-Group/AdmixSim2.

2. Getting Started

For Cpp version, the boost libraries (http://www.boost.org) are required. The executable file is already included in the src folder.

```
$ tar -zxvf AdmixSim2.tar.gz
$ cd AdmixSim2/src
```

To recompile from the source code, you can type the following commands:

```
$ make clean
$ make
```

Then the executable file can be found in the src folder, and you can type the following command for help:

```
$ ./AdmixSim2 -h
```

Test with toy data (can be found in folder example1)

```
$ ./AdmixSim2 -in test -p Admixed -g 8 -n 4 -mut 0.00000001 -out out1
```

For Python version, the python version 3 and package numpy are required. You can type the following command for help:

```
$ python AdmixSim2.py -h
```

Test with toy data

```
$ python AdmixSim2.py -i test -p Admixed -g 8 -n 4 --mut-rate 0.00000001 -o out1
```

Explanation:
In the example data, there are four ancestral populations, namely Anc1, Anc2, Anc3, and Anc4, and three new admixed populations, namely Admixed1, Admixed2, and Admixed. From generation 1 to 4, Admixed1 is descendants of Anc1 and Anc2, while Admixed2 is descendants of Anc3 and Anc4. From generation 5 to 8, Admixed is descendants of Admixed1 and Admixed2. In the simulation process, the uniform mutation rate is 10^{-8} per generation per site (`-mut 0.00000001`) and the genetic distance (Morgan) is locus-specific which is illustrated in the input single nucleotide variation (SNV) information file (.snv). There is no selection event in this model. At the end of simulation, four individuals (`-n 4`) of population Admixed (`-p Admixed`) at generation eight (`-g 8`) are sampled out without replacement. Six output files (.ind, .hap, .snv, .seg, .sel, and .log) are named with the prefix "out1" (`-out out1`). Details of each file are explained in the section 4 and 5.

3. Arguments and Options

For Cpp version, the arguments and options are as follows:

Arguments	Type	Description	Note
-in/--inPrefix	string	The prefix of four input files	required
-mod/--modfile	string	Model description file	required
-ind/--indfile	string	Individual information file	required
-hap/--hapfile	string	Ancestral haplotype file	required
-snv/--snvfile	string	SNV information file	required
-p/--poplist	string	Output population list, separated by comma	optional, default = the final generated population

Arguments	Type	Description	Note
-g/--gen	integer	The generation index of each output population, separated by comma	optional, default = the last generation in simulation
-n/--nInd	integer	The sampled number of each output population, separated by comma	optional, default = 10
-s/--seed	integer	Seed of random generation	optional, default = time
-c/--chr	string	Type of chromosome to simulate: A (autosome), X (X-chromosome)	optional, default = A
-rec/--recRate	double	The uniform recombination rate for all loci (unit: Morgan per base pair)	optional
-mut/--mutRate	double	The uniform mutation rate for all loci (unit: per generation per site)	optional
-out/--outPrefix	double	The prefix of all output files	optional, default = out

Options	Description	Note
--no-rec	No recombination events in the simulation	optional
--no-mut	No mutation events in the simulation	optional
--no-sel	No selection events in the simulation	optional
-h/--help	Print help message	optional

Additional explanations:

1. When the prefixes of four input files are same, you can use the argument `-in/--inPrefix` to simplify your command line. Arguments `-mod/--modfile`, `-snv/--snvfile`, `-hap/--hapfile`, and `-ind/--indfile` have higher priority than `-in/--inPrefix`.
2. The argument `-out/--outPrefix` can be the directory of ouput file, or flags to distinguish different tasks.

For Python version, the arguments and options are as follows:

Arguments	Type	Description	Note
-i	string	The prefix of all input files	required
--mod	string	The model file describing admixture events and selection events	required

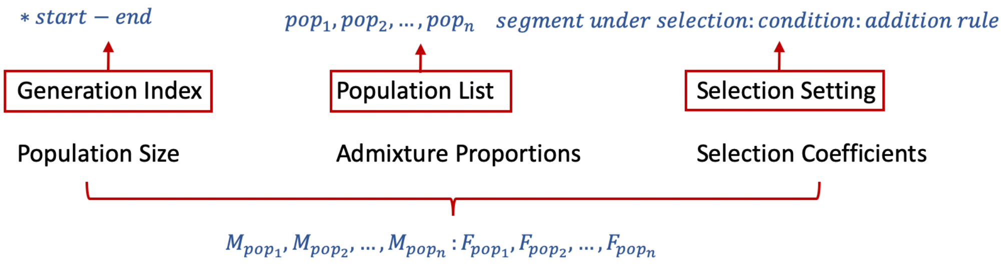
Arguments	Type	Description	Note
--ind	string	The information of population and gender of n ancestral individuals	required
--hap	string	The haplotype file containing $2N$ haplotypes for N ancestral individuals	required
--snv	string	The SNV file providing a genetic distance and a mutation rate of simulated each site	required
-p	string	Names of populations for output	optional, default = the final generated population
-g	integer	Generations of population corresponding to `-p` for output	optional, default = the last generation in the modfile
-n	integer	Numbers of populations corresponding to `-p` for output	optional, default = 10
--chr	string	Set the type of simulation genomes. 'A' for autosomes, 'X' for X chromosome.	optional, default = A
--rec-rate	string	A uniform recombination rate across the simulated genome while ignoring genetic distance in the SNV file	optional
--mut-rate	string	A uniform mutation rate across the simulated genome while ignoring mutation rates in the SNV file	optional
-o/--out	string	The prefix of output files	optional, default = out

Options	Description	Note
--no-rec	No recombination in the simulation process	optional
--no-mut	No mutation in the simulation process	optional
--no-sel	No selection in the simulation process	optional
-h/--help	Print help message	optional

4. Input Files

4.1 Model Description File (.mod)

In model description file, each module can mainly be divided into six parts, shown in the following figure:


$* start - end$
Generation Index
$pop_1, pop_2, \ldots, pop_n$
Population List
$segment\ under\ selection: condition: addition\ rule$
Selection Setting
Population Size
Admixture Proportions
Selection Coefficients
$M_{pop_1}, M_{pop_2}, \ldots, M_{pop_n} : F_{pop_1}, F_{pop_2}, \ldots, F_{pop_n}$

In each module, you can specify the start, end generation index, and populations involved. If selection events are allowed in the module, you can set the physical position(s) under selection, corresponding selection conditions, and addition rule in the first line, starting from the third column. From the second line, you can specify the population size and mixture proportions of each population. With selection events, you can set up the selection coefficients in the corresponding column from the second line for each generation. Population size, admixture proportions, and selection coefficients take the same format as $M_{pop_1}, M_{pop_2}, \ldots, M_{pop_n} : F_{pop_1}, F_{pop_2}, \ldots, F_{pop_n}$. Here, M represents male and F represents female. For more complicate simulation model, you can combine multiple modules to illustrate the simulation process. Below is the detailed explanation.

Firstly, start with the asterisk "*", and then set up the start, end generation index, and populations involved. Here, use the comma "," to separate different populations. Note that there should be no space after comma. If a new admixed population emerged in the module, you need to put it at the end of involved population list. Note that for simulation with new admixed population, the number of ancestral populations must be greater or equal to two. In each module, we do not support events of introgression but no new admixed population generated. This means that in a module, if no new admixed population is pointed out in the first line, the following admixture proportion settings are treated as '1' and each population undergoes self-evolution. Moreover, the number of new admixed population in each module can only be one or zero. In the following example, the generation index is from 1 to 5, and the ancestral populations are named as Anc1 and Anc2. The Admixed1 is a new admixed population in this module.

```
*1-5    Anc1,Anc2,Admixed1
```

If selection events are allowed, you can set the physical position(s) under selection, corresponding selection conditions, along with addition rules in the first line, using the colon ":" to separate. Note that there should be no space after colon. The physical position under selection must be in the initial SNV information file (.snv), cannot be the new mutation sites. For addition rule, the optional values are 1 (additive model, 1, 1+s, 1+2s for individuals carrying 0, 1, 2 selected haplotypes) and 2 (dominant model, 1, 1+s, 1+s for individuals carrying 0, 1, 2 selected haplotypes). The default value is

1 and fitness is addable for each individual. In the following example, allele 0 at locus 16469059 is under selection, and the addition rule is additive model:

```
*1-5  Anc1,Anc2,Admixed1  16469059:0:1
```

Secondly, set up the population size. Here, you can set up population-specific size for each population, separated by comma ",". Use colon ":" to separate different settings for two sexes. Note that there should be no space after comma and colon. In the next column, set up mixture proportions. If there is a new admixed population in the module, the mixture proportions need to be set specially for the admixed one and the sum of mixture proportions should be 1. Otherwise, the mixture proportion needs to be set as 1. If selection events are allowed, you can set up the selection coefficients starting from the third column for each generation, same format as settings for population sizes and admixture proportions. Furthermore, if you want to perform the negative selection, the selection coefficient must not be smaller than -1. When generating each offspring in the next generation, *AdmixSim 2* randomly samples two different individuals in the current generation as parents proportionally to their fitness. The fitness is computed based on selection coefficients and addition rules specified by user. Negative value is rescaled to zero.

In the following example, male population sizes for Anc1, Anc2, and Admixed1 in the first generation are 60, 60, and 50. Correspondingly, female population sizes for each population are 40, 40, and 50. The admixed proportions for Admixed1 are 10% from Anc1 males, 40% from Anc2 males, 30% from Anc1 females, and 20% from Anc2 females.

```
*1-5                Anc1,Anc2,Admixed1    16469059:0:1
60,60,50:40,40,50   0.1,0.4,0:0.3,0.2,0   1,1,1:1.2,1.2,1.2
```

For sex-specific admixture proportions setting, if the sum of male proportions does not equal to that of female proportions, the simulation will automatically rescale the proportions to ensure that the sum of male proportions equals to that of female proportions, which is more reasonable in practice.

If you do not want to set different values for two sexes, you can set the total population size for each population. The same goes for admixture proportions and selection coefficients.

```
*1-5                Anc1,Anc2,Admixed1    16469059:0:1
60,60,50:40,40,50   0.1,0.4,0:0.3,0.2,0   1,1,1:1.2,1.2,1.2
100,100,100         0,0,1                 1,1,1
```

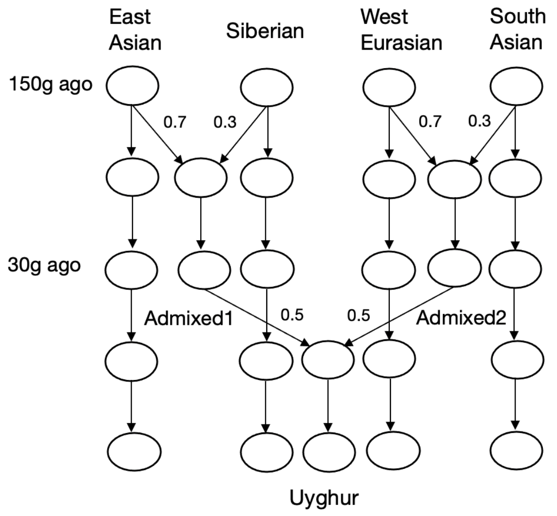
Furthermore, if the settings are same for each population, you can use one value for all populations.

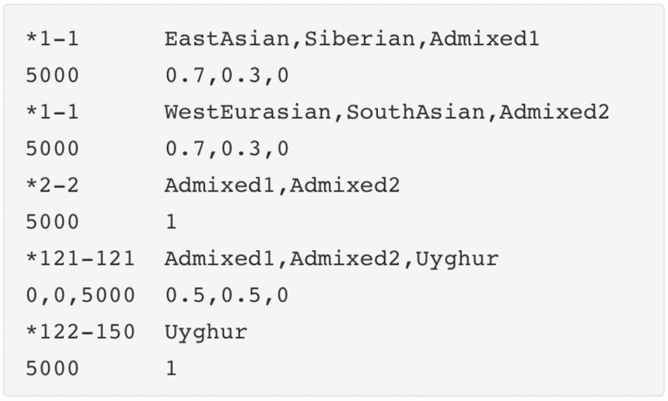
```
*1-5                Anc1,Anc2,Admixed1     16469059:0:1
60,60,50:40,40,50   0.1,0.4,0:0.3,0.2,0    1,1,1:1.2,1.2,1.2
100,100,100         0,0,1                  1,1,1
100                 0,0,1                  1
100                 0,0,1                  1
100                 0,0,1                  1
```

Note that settings from generation 2 to generation 5 are same. For simplicity, you can omit generations that have same settings as the last generation, including population sizes, admixture proportions, and selection settings. The program complement the simulation process automatically. In detail, the end generation index of each population is the maximum of corresponding model settings and sampled generation index specified by parameter `-g/--gen`. The above admixture model can be simplified as follows:

```
*1-2                Anc1,Anc2,Admixed1     16469059:0:1
60,60,50:40,40,50   0.1,0.4,0:0.3,0.2,0    1,1,1:1.2,1.2,1.2
100                 0,0,1                  1
```

As mentioned before, you can combine multiple modules to illustrate more complex admixture process. Here we use the simplified admixture model of population Uyghur as an example:

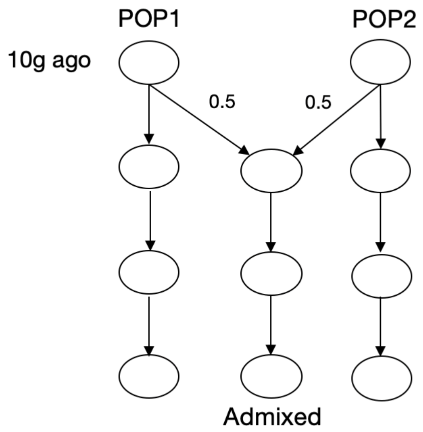


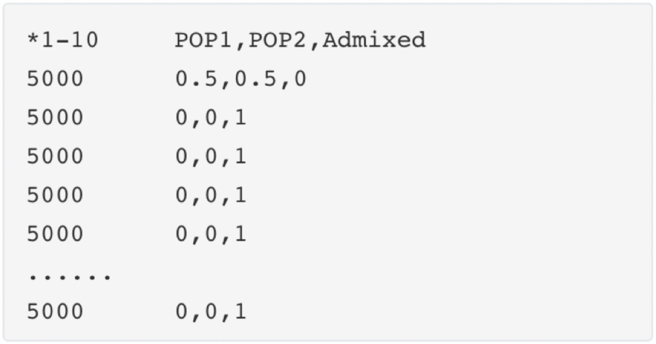


```
*1-1       EastAsian,Siberian,Admixed1
5000       0.7,0.3,0
*1-1       WestEurasian,SouthAsian,Admixed2
5000       0.7,0.3,0
*2-2       Admixed1,Admixed2
5000       1
*121-121   Admixed1,Admixed2,Uyghur
0,0,5000   0.5,0.5,0
*122-150   Uyghur
5000       1
```

By the same manner, it is quite simple to implement HI, GA, CGFR, or CGFD models:

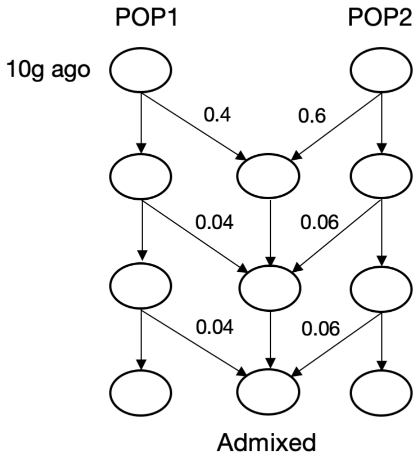
HI model:

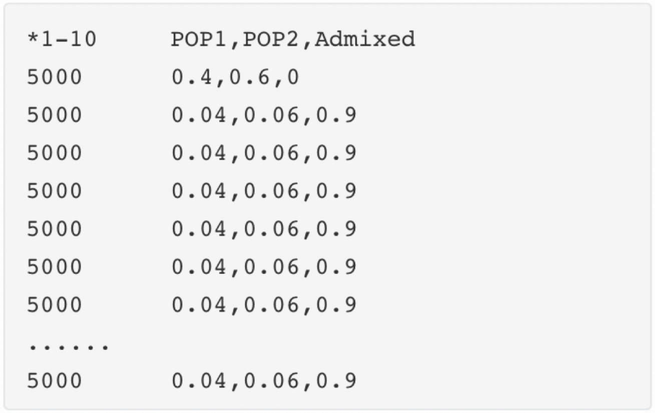




```
*1-10     POP1,POP2,Admixed
5000      0.5,0.5,0
5000      0,0,1
5000      0,0,1
5000      0,0,1
5000      0,0,1
......
5000      0,0,1
```

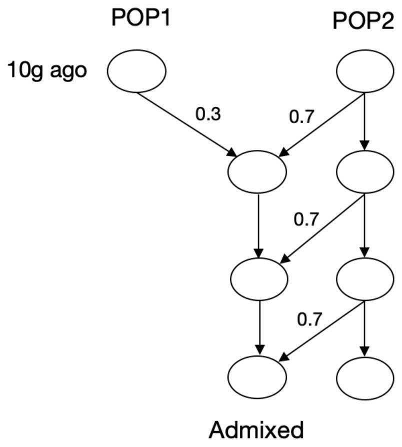
GA model:

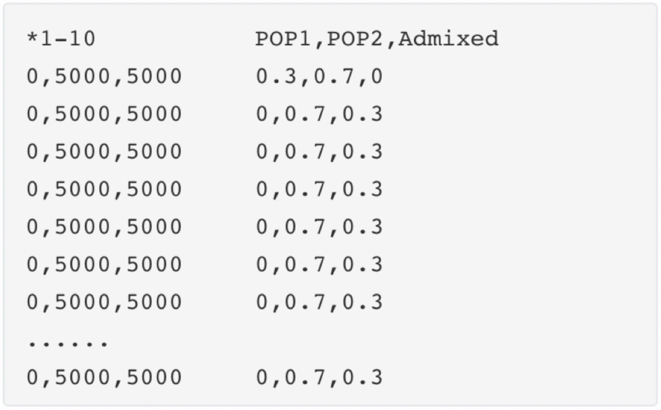




```
*1-10     POP1,POP2,Admixed
5000      0.4,0.6,0
5000      0.04,0.06,0.9
5000      0.04,0.06,0.9
5000      0.04,0.06,0.9
5000      0.04,0.06,0.9
5000      0.04,0.06,0.9
5000      0.04,0.06,0.9
......
5000      0.04,0.06,0.9
```

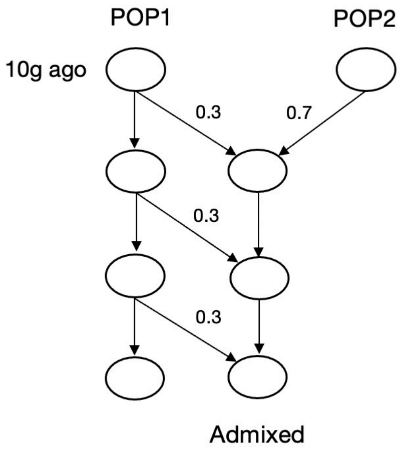
CGFR model:





```
*1-10            POP1,POP2,Admixed
0,5000,5000      0.3,0.7,0
0,5000,5000      0,0.7,0.3
0,5000,5000      0,0.7,0.3
0,5000,5000      0,0.7,0.3
0,5000,5000      0,0.7,0.3
0,5000,5000      0,0.7,0.3
0,5000,5000      0,0.7,0.3
......
0,5000,5000      0,0.7,0.3
```

CGFD model:



```
*1-10           POP1,POP2,Admixed
5000,0,5000     0.3,0.7,0
5000,0,5000     0.3,0,0.7
5000,0,5000     0.3,0,0.7
5000,0,5000     0.3,0,0.7
5000,0,5000     0.3,0,0.7
5000,0,5000     0.3,0,0.7
5000,0,5000     0.3,0,0.7
......
5000,0,5000     0.3,0,0.7
```

In addition to admixture, we also support the process of no new admixed population emerging and each population undergoing self-evolution, for example,

```
*1-10   POP1,POP2
5000    1
5000    1
......
5000    1
```

Settings in model description file can be summarized by the following four principles:

1. Use comma to separate settings for different populations.
2. Use colon to separate settings for two sexes.
3. Combine multiple modules for complex admixture model.
4. Omit same generation settings for simplicity.

Furthermore, we want to introduce some other ways to set up the selection events. Suppose the input physical positions in SNV information file are as follows:

```
10 20 30 40 50 60 70
```

The conditions under selection can be combination of alleles or ancestry-specific. In the latter case, the ancestry under selection can only be the initial input population, which is recorded in the individual information file (.ind).

```
*1-2  Anc1,Anc2,Admixed1  60:Anc1
```

If the segment under selection is continuous, you can use the "-" to connect the start and end physical positions of this segment.

```
*1-2  Anc1,Anc2,Admixed1  40-60:001
```

If the segment under selection is discrete, you can use the comma "," between physical positions.

```
*1-2  Anc1,Anc2,Admixed1  50,70:01
```

Or you can combine these two ways (The Python version does not support this feature for now).

```
*1-2  Anc1,Anc2,Admixed1  20-40,60:0001
```

Besides, the condition under selection can be multiple combinations of alleles. Also, use the comma "," to separate different allele combinations and no space after comma.

```
*1-2  Anc1,Anc2,Admixed1  40,60:01,00
10    0.5,0.5,0           1
10    0.4,0,0.6           1
```

Note that if the selection coefficients of different allele combinations are different, you need to separate them into two columns.

```
*1-2  Anc1,Anc2,Admixed1  40,60:01  40,60:00
10    0.5,0.5,0           1         2
10    0.4,0,0.6           1         2
```

Moreover, if you don't want to simulate selection events, you can use the hash "#" to comment out the selection settings or delete them in the model description file. Any contents after "#" are treated as comments and ignored. For example:

```
*1-2  Anc1,Anc2,Admixed1  #40,60:01
10    0.5,0.5,0           #1
10    0.4,0,0.6           #1
```

Or you can just use the option `--no-sel` to avoid selection events without modifying your model description file.

4.2 Individual Information File (.ind)

In individual information file, you can specify the individual ID, source population label, and sex of each individual. Note that individual ID cannot be repeated. For sex setting, 1 means male, 2 means female, and 0 means unknown. The order of individual information must be same as that in haplotype data file. Here is an example:

```
Ind1    Anc1    1
Ind2    Anc1    2
Ind3    Anc1    1
Ind4    Anc2    2
......
```

In detail, we can simulate the situation of monoecious and dioecious individuals according to the initial individual sex settings. In the case of monoecious, we logically sample two different individuals to be parents of each offspring in the next generation. The sex of each offspring is denoted as 0. Under this circumstances, you cannot perform the sex-specific simulation. Otherwise, if the input individuals are dioecious, the sex of each individual in the simulation is denoted as 1 or 2, and sex-specific simulations are allowed.

Note that for X chromosome simulation, you must specify the clear sex of each input individual. Otherwise, the program will report an error and then terminate the simulation. Furthermore, if the sex of each individual in a population is specified as 1 or 2 and it needs to self-evolve at a certain generation, there must be at least one male and one female in this population. Otherwise, it cannot produce offsprings and the program will report an error.

4.3 Haplotype Data File (.hap)

Haplotype data file contains one haplotype data per line, and each column corresponds to a locus in the SNV information file. The character can be in binary format: '0' denotes the reference allele and '1' denotes the alternative allele. Here is an example:

```
0000000000000100010001100000110101101001000000000000000100111000
0100001000100100001100001000000000000000000000001000000110000000
......
0011000001100000000001100000010000000000000000010101110000000000
```

The character can also be a DNA base. For example:

```
AGCTTAGCAGATAGATCGGACGATGATTAGCAGATAGATCGGACGATGATTAGCAGATAGATCG
CTTAGCAGATAGATCGGACGATGATTAGCAGATAGATCGGACGATGATTAGCAGATAGATCGGA
......
CGTTTCACATATGTCTGGCAATATGTTTAAGGTTTGGATGGCGAGGATTTGAGGGTTGGTATGG
```

If you want to perform the X chromosome simulation, you can use "9" in the second line for male individuals.

```
0100000100111000000000001101000000110000000000000000000000000001
9999999999999999999999999999999999999999999999999999999999999999
```

4.4 SNV Information File (.snv)

SNV information file specifies the physical position, genetic distance, and mutation rate of each locus, one locus per line. The unit of each column are in base pair, Morgan, and per generation per site, respectively. Note that physical position and genetic distance, if specified, must be in strictly ascending order. Here is an example:

```
16469059        0.02114202      0.000000010
19006125        0.09413300      0.000000011
20936109        0.13327470      0.000000012
22922471        0.18263998      0.000000011
24447089        0.22590243      0.000000012
26213574        0.25469293      0.000000010
27623435        0.30662273      0.000000012
29456671        0.34093637      0.000000012
......
```

If you don't want to specify the genetic distance or mutation rate for each locus, you can use the parameter `-rec/--recRate` (Cpp), `--rec-rate` (Python) or `-mut/--mutRate` (Cpp), `--mut-rate` (Python) to set up the uniform recombination rate (Morgan per base pair) or mutation rate (per generation per site) and then use "-9" in the corresponding column of SNV information file. For example:

```
16469059        0.02114202      -9
19006125        0.09413300      -9
20936109        0.13327470      -9
22922471        0.18263998      -9
24447089        0.22590243      -9
26213574        0.25469293      -9
27623435        0.30662273      -9
29456671        0.34093637      -9
......
```

The uniform recombination rate (or mutation rate) has higher priority than locus-specific genetic distance (or mutation rate).

5. Output Files

After simulation, there are six output files recording different information. They are individual information file (.ind), haplotype data file (.hap), updated SNV information file (.snv), ancestral track file (.seg), selection frequency file (.sel), and log file (.log).

5.1 Individual information File (.ind)

At the end of simulation, N individuals (`-n/--nInd`) are randomly sampled out without replacement from the specified populations (`-p/--poplist`) at the specified generation (`-g/--gen`). This file records the information of output individuals. The format is same as input individual information file described above. Population label in the second column is named after population name and generation index, using '_' to connect.

5.2 Haplotype Data File (.hap)

This file records the haplotype data of output individuals. The format is same as input haplotype data file described above.

5.3 Updated SNV Information File (.snv)

This file records the updated SNV information of output individuals. The first three columns have the same format as input SNV information file. Last column annotates whether the locus is de novo mutation ('T') or not ('F'). For each de novo mutation, genetic distance in the second column is calculated using linear interpolation and mutation rate in the third column is same as the the mutation rate of minimum physical position greater than or equal to the corresponding mapped physical position. In the last column, reference and alternative allele are also be demonstrated for each mutation site. If the input sequence data is in binary format, the reference allele is '0' and the alternative allele is '1'. Otherwise, if the data is DNA base, the reference allele is a random one in 'A','G','C','T', and the alternative allele is a random one in the other three. Here are two examples:

```
16469059        0.02114202      1e-08   F
19006125        0.094133        1e-08   F
20936109        0.1332747       1e-08   F
22922471        0.18263998      1e-08   F
23169236        0.1896421658    1e-08   T,0/1
24196684        0.2187969558    1e-08   T,0/1
24447089        0.22590243      1e-08   F
26213574        0.25469293      1e-08   F
......
```

and

```
16469059        0.02114202      1e-08   F
19006125        0.094133        1e-08   F
20936109        0.1332747       1e-08   F
22922471        0.18263998      1e-08   F
23169236        0.1896421658    1e-08   T,A/G
24196684        0.2187969558    1e-08   T,C/G
24447089        0.22590243      1e-08   F
26213574        0.25469293      1e-08   F
......
```

5.4 Ancestral Tracks Information File (.seg)

This file specifies the physical position (base pair) of start point, genetic distance (Morgan) of start point , physical position (base pair) of end point, genetic distance (Morgan) of end point, and from which population the track originates from. Segments from the same ancestry are merged. The ancestral tracks information file records the ancestral origin of each segment, which facilitates users estimating the length distribution of ancestral tracks, identifying the segments of identity by descent, and investigating the recombination patterns. Notes that only tracks from output individuals are saved. Besides, when simulating X chromosomal data, only the first haplotype segment information of males is recorded. For example:

```
Ind1 Hap 1
16469059        0.02114202      23976120        0.2125382477    Anc1
23976120        0.2125382477    47601123        0.6243576526    Anc2
47601123        0.6243576526    48849346        0.66376859      Anc1
Ind1 Hap 2
16469059        0.02114202      36275930        0.4250721252    Anc4
36275930        0.4250721252    37634829        0.455307771     Anc2
37634829        0.455307771     46661084        0.6041259271    Anc4
46661084        0.6041259271    47601123        0.6243576526    Anc2
47601123        0.6243576526    48849346        0.66376859      Anc1
......
```

5.5 Selection Frequency File (.sel)

This file contains the frequencies of loci under selection of each population at each generation. The title of each column are generation, population, physical position(s) under selection, corresponding condition, frequency, male frequency, and female frequency. If the sexes of input individuals are '0', the Male_Frequency and Female_Frequency are marked as '-'. The following output file can be found in the folder example2:

```
Gen   Pop       Position(s)  Condition Frequency  Male_Frequency Female_Frequency
1     Admixed1  16469059     0         0.9        1              0.833333
1     Admixed2  20936109     1         0.6        0.75           0.5
1     Anc1      16469059     0         1          1              1
1     Anc2      16469059     0         1          1              1
1     Anc3      20936109     1         0.5        0.5            0.5
1     Anc4      20936109     1         1          1              1
2     Admixed1  16469059     0         1          1              1
2     Admixed2  20936109     1         0.8        0.75           0.833333
3     Admixed1  16469059     0         1          1              1
3     Admixed2  20936109     1         0.8        0.75           0.833333
......
```

If there is no selection event in the simulation process, this file only have the header line.

```
Gen   Pop       Position(s)  Condition Frequency  Male_Frequency Female_Frequency
```

With output file of allele frequency of variants underlying selection, users can further monitor the fluctuation of frequencies across populations and generations, which is helpful in estimating the fixation time of a particular allele under different selection conditions.

5.6 Log File (.log)

This file records the input filename, parameter and argument settings of simulation. Besides, it records the start time, end time, and consuming time of each critical part during the simulation. Here is an example:

```
AdmixSim 2
Arguments and Options:
  Modfile = test.mod
  SNVfile = test.snv
  Hapfile = test.hap
  Indfile = test.ind
  Output population list and corresponding generation and population size:
Admixed 8g 4;
  Seed = 1599362113
  Chr = A
  Recombinaion Setting: Using locus-specific genetic distance in snv file
  Mutation Setting: Uniform mutation rate = 1e-08
  Out prefix = out1
=================================================================================
=========================

Simulation start! Sun Sep  6 11:15:13 2020

Reading snvfile time: 0s
Reading indfile time: 0s
Reading modfile time: 0s
Reading hapfile time: 0s

Simulation time: 0s

Saving ancestral sequence data time: 0s

Final add de novo mutation and output snvfile time: 0s

Population Admixed at generation 8 output time: 0s

Simulation end! Sun Sep  6 11:15:13 2020
```

6. Examples

Here we give some simple examples to illustrate basic features of our simulation tool. Detailed information can be found in the following table:

Feature	Description
admixture	one or more new admixed populations
recombination	locus-specific/ constant/ absent
mutation	locus-specific/ constant/ absent
natural selection	single locus/ multi-locus/ population-specific/ sex-specific/ absent
chromosome	autosomes/ sex chromosomes

6.1 Recombination

For recombination, you can use the argument `-rec/--recRate` (Cpp), `--rec-rate` (Python) to set up the uniform recombination rate for all loci.

```
$ ./AdmixSim2 -in test -p Admixed -g 8 -n 4 -rec 0.00000002 -mut 0.00000001 -out out1
```

Or you can specify the genetic distance for each locus in the second column of SNV information file (.snv).

```
16469059        0.02114202      -9
19006125        0.09413300      -9
20936109        0.13327470      -9
22922471        0.18263998      -9
24447089        0.22590243      -9
......
```

The number of recombination breakpoints per chromosome is drawn from a Poisson distribution with the parameter as genetic distance of last locus.

If you do not want to simulate recombination events, you can realize it using the option `--no-rec`.

```
$ ./AdmixSim2 -in test -p Admixed -g 8 -n 4 --no-rec -mut 0.00000001 -out out1
```

6.2 Mutation

For mutation, you can use the argument `-mut/--mutRate` (Cpp), `--mut-rate` (Python) to set up the uniform mutation rate for all loci.

```
$ ./AdmixSim2 -in test -p Admixed -g 8 -n 4 -mut 0.00000001 -out out1
```

Or you can specify the mutation rate for each locus in the third column of SNV information file (.snv).

```
16469059        0.02114202      0.000000010
19006125        0.09413300      0.000000011
20936109        0.13327470      0.000000012
22922471        0.18263998      0.000000011
24447089        0.22590243      0.000000012
......
```

The number of mutaion sites per chromosome is drawn from a Poisson distrbution with the parameter as defined mutation distance of last locus. Here, the caculation method of defined mutation distance is similar to the genetic distance. New generated mutations are then mapped to a new physical position using linear interpolation.

If you do not want to simulate the situation of mutation events, you can use the option `--no-mut` to turn off.

```
$ ./AdmixSim2 -in test -p Admixed -g 8 -n 4 --no-mut -out out1
```

6.3 Natural Selection

For selection, we provide various ways to illustrate the conditions under selection. The detailed information can be found in the section 4.1. The following command line is for simulation data in folder example2.

```
$ ./AdmixSim2 -in test_select -p Admixed -g 8 -n 4 -out out2
```

If you do not want to simulate the situation of selection events, you can use the option `--no-sel` to turn off.

```
$ ./AdmixSim2 -in test_select -p Admixed -g 8 -n 4 --no-sel -out out2
```

6.4 X chromosome simulation

The differences of input file formats between X chromosome and autosome simulation can be summarized into following two parts:

1.For male individuals in the haplotype data file (.hap), you need to use "9" in the second line.

```
010000010011100000000000110100000011000000000000000000000000001
999999999999999999999999999999999999999999999999999999999999999
```

2.In the individual information file (.ind), '0' is forbidden for sex setting in X chromosomal data simulation. You need to specify the clear sex for each individual.

```
ind1    Anc1    1
ind2    Anc1    2
ind3    Anc1    1
......
```

In X chromosome simulation, recombination events only happen in females. Here we provide an example for X chromosomal data simulation, which can be found in the folder example3. The corresponding command line is:

```
$ ./AdmixSim2 -in test.X -p Admixed -g 8 -n 4 -c X -out out3
```

7. Questions and Suggestions

Questions and suggestions are welcome. Feel free to contact zhangrui2018@picb.ac.cn or liuchang@picb.ac.cn.