User Manual

Introduction

This pipeline provides an automatic way to run simulations through simuPOP once or multiple times. It has been designed specifically for the purposes of this project, but the pipeline could also be used for a more general purpose if you change the parameters and if needed the code. The code provided simulates an island model with 4 populations. For a different population structure, you need to change the Migrator operator in the model.py script. For further information on this please advise the User Guide of simuPOP. Once you have the data (haplotypes), you can easily convert them to the format you want to run the selection methods.

Requirements

To run the pipeline, you need to install:

- ipython and pylab which will provide you with many python packages automatically.
- SimuPOP: the easiest way to install simuPOP in any machine is to download the binary version and put it in the python packages.

Run the pipeline

In a terminal, you should type the following:

```
ipython --pylab
import models as s
```

To run just one simulation:

```
g = s.Model(Gen=200,loci=101,dist=4,alleles=2,numPop=4)

g.run(sizePop=200,step=2,recombination_rate=0.00375,migration_rate=0.01,s1=0
.1,mutation_rate=0.00000001,subPopNames = ['x','y','z','w'],burnin=10)
```

where:

Gen= number of generations

Loci= number of loci

Alleles = number of alleles

numPop = number of populations

sizePOP = subpopulation size

step = step between generations

recombination_rate = recombination rate

migration_rate = migration rate

s1 = selection coefficient

mutation_rate = mutation rate

subPopNames = names of subpopulations. In the case you want to have a population structure with more than 4 subpopulations and you want to save the results, you need to add some additional code.

Output

In the end of the simulation you will have a dictionary with all the results.
The data are the following:

1. All allele frequencies of all loci over all generations in the population which is under selection
2. All the haplotype for all subpopulations for the different allele frequencies of the selected locus (0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)
3. The allele frequency of locus 50 over all the generations for all the subpopulations
4. Fst calculation based on all loci over the generations of the simulation

Description of the output

(1) `g.results.allelesFreq`

if you want to access the allele frequencies of a specific loci:

```
        g.results.allelesFreq['alleleFr500']
```

where 50 is the locus you want to check and 0 will provide you with the allele frequencies of allele 0

(2) `g.results.all_haplotypes`

if you want to access the haplotypes of a specific population:

```
        g.results.all_haplotypes['y08'])
```

where y is the population and 08 indicate the you have the haplotypes when the selected locus is close to the allele frequency of 0.8.

In this way you can access all the haplotypes from any subpopulation.

In the specific example described in the test file, we have 200 haplotypes from 100 individuals with 101 loci each.

(3) Allele frequency of locus 50 over all the generations for the subpopulation Y

```
g.results.YSelectedLoci
```

Allele frequency of locus 50 over all the generations for the subpopulation Z

```
g.results.ZSelectedLoci
```

Allele frequency of locus 50 over all the generations for the subpopulation W

```
g.results.WSelectedLoci
```

Allele frequency of locus 50 over all the generations for the subpopulation X

```
g.results.XSelectedLoci
```

(4) `g.results.fst`

To run multiple simulations:

The procedure is the same with the following differences:

- use MultiModel instead of Model
- add the number of simulation you want
- all the results are save in the object multi (dictionary type)

Example:

```
g = s.MultiModel(Gen=200,loci=101,dist=4,alleles=2,Nruns=2,numPop=4)

g.run(sizePop=200,step=2,recombination_rate=0.00375,migration_rate=0.01,s1=0
.1,mutation_rate=0.00000001,subPopNames = ['x','y','z','w'],burnin=10)
```

Output

All the outputs are lists that contain all the results of all the simulations. That means that the length of all of the following outputs will be equal to the number of simulation that you defined.

```
g.multi.all_Sim_alleleFreq
g.multi.all_Sim_ haplotypes
g.multi.all_YSelectedLoci
g.multi.all_WSelectedLoci
g.multi.all_ZSelectedLoci
g.multi.all_XSelectedLoci
g.multi.all_fst
```

Please feel free to contact me in alex.vatsiou@gmail.com if you want to run simulations and you have difficulties.