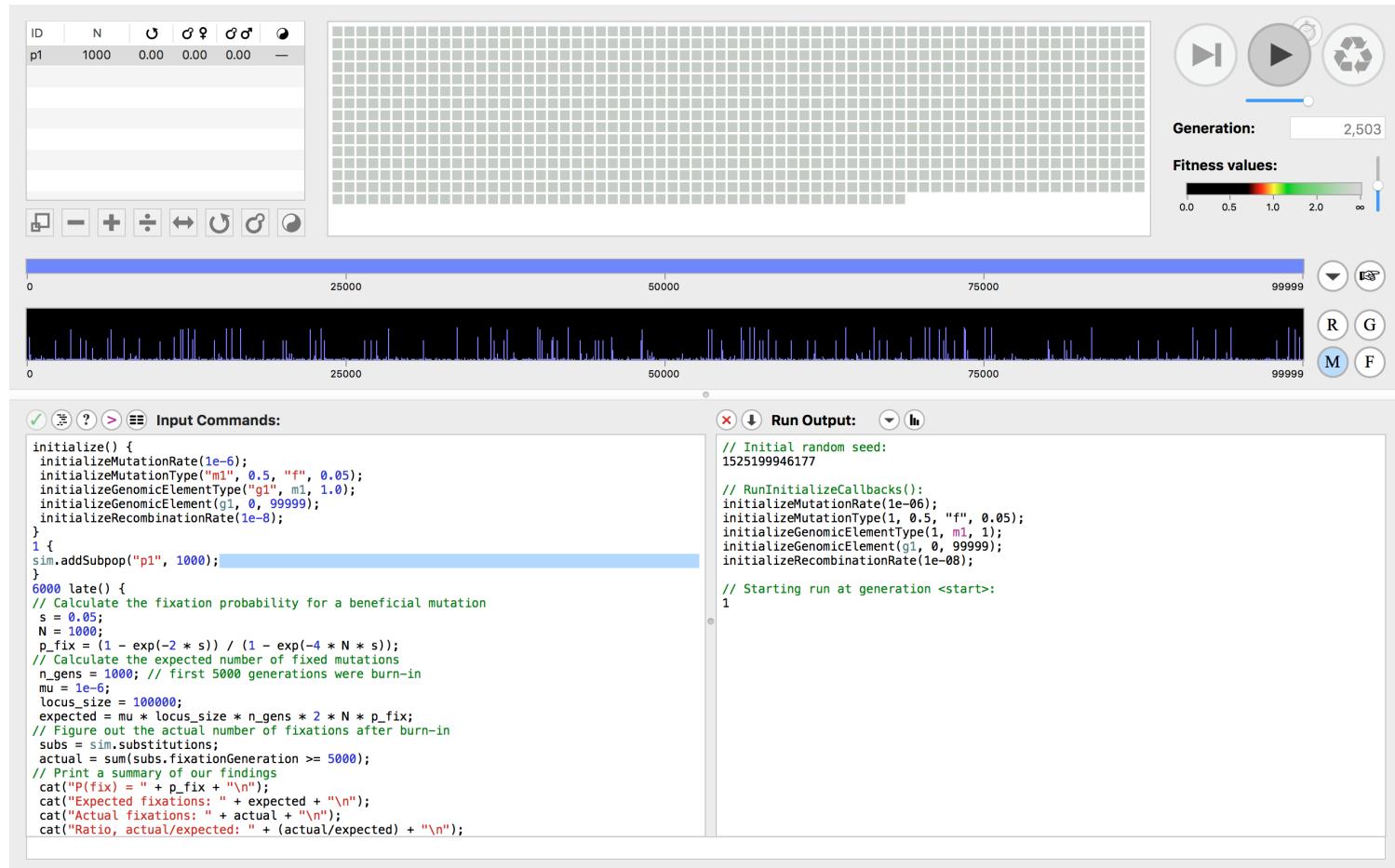


An introduction to evolutionary simulations with SLiM



Ethan Linck
Taylor Lab meeting, CU Boulder, 091819

Why simulate?

...to answer tricky biological questions

Felsenstein, J. 1981. Skepticism towards santa rosalia, or why are there so few kinds of animals? *Evolution* 35:124-138. doi: <https://doi.org/10.1111/j.1558-5646.1981.tb04864.x>

Kirkpatrick, M. 1997. Evolution of a Species' Range. *The American Naturalist* 150: 1-23. doi: <https://doi.org/10.1086/286054>

Battey, C.J., Ralph, P., Kern., A.D. 2019. Space is the Place: Effects of Continuous Spatial Structure on Analysis of Population Genetic Data. bioRxiv: 659235. doi: <https://doi.org/10.1101/659235>

Linck, E., Battey, C.J. 2019. On the relative ease of speciation with periodic gene flow. bioRxiv: 758664. doi: <https://doi.org/10.1101/758664>

...to perform power analyses / test tools

Cornuet, J.M., Luikart, G. 1996. Description and Power Analysis of Two Tests for Detecting Recent Population Bottlenecks From Allele Frequency Data. *Genetics* 144:2001-2014. url:
<https://www.genetics.org/content/144/4/2001>

Luikart, G., Allendorf, FW, Cornuet, J-M, Sherwin, WB. 1998. Distortion of allele frequency distributions provides a test for recent population bottlenecks. *Heredity* 89: 238-247. doi:
<https://doi.org/10.1093/jhered/89.3.238>

Ramírez-Soriano, A., Ramos-Onsins, S.E., Rozas, J., Calafell, F., Navarro, A. 2008. Statistical Power Analysis of Neutrality Tests Under Demographic Expansions, Contractions and Bottlenecks With Recombination. *Genetics* 179: 555-567. doi:
<https://doi.org/10.1534/genetics.107.083006>

Linck, E., Battey, C.J. 2019. Minor allele frequency thresholds strongly affect population structure inference with genomic datasets. *Molecular Ecology Resources* 19: 639-647. doi: <https://doi.org/10.1111/1755-0998.12995>

...for demographic inference

Beaumont, M.A., Zhang, W., Balding, D. 2002. Approximate Bayesian Computation in Population Genetics. *Genetics* 162: 2025-2035. url:
<https://www.genetics.org/content/162/4/2025.long>

Nicolaisen, L.E., Desai, M.M. 2012. Distortions in Genealogies Due to Purifying Selection. *Molecular Biology and Evolution* 29: 3589–3600

Gladstein, A., Hammer, M.F. 2019. Substructured Population Growth in the Ashkenazi Jews Inferred with Approximate Bayesian Computation. *Molecular Biology and Evolution* 36: 1162–1171. doi:
<https://doi.org/10.1093/molbev/msz047>

Linck, E., Freeman, B.G., Dumbacher, J.P.D. 2019. Speciation with gene flow across an elevational gradient in New Guinea kingfishers. *bioRxiv*: 589044. doi: <https://doi.org/10.1101/589044>

Coalescent simulations

Advantages:

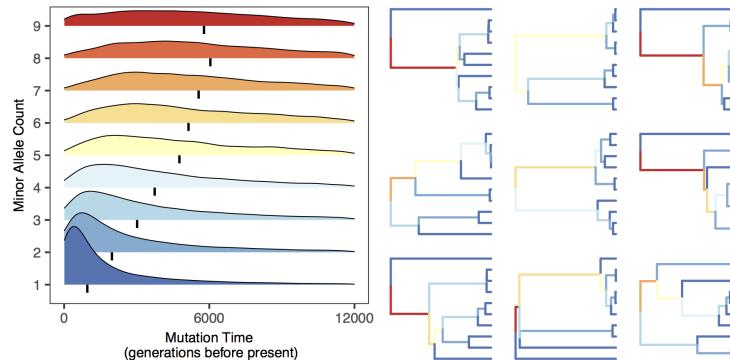
- very fast
- backed by lots of theory
- many tools available

Disadvantages:

- difficult to incorporate selection
- other realistic quirks of biology even harder

Examples:

- msprime (<https://msprime.readthedocs.io/en/stable/>)
- fastsimcoal2 (<http://cmpg.unibe.ch/software/fastsimcoal2/>)
- coalesceR (<https://github.com/cjbattey/coalesceR>)



Forward time simulations

Advantages:

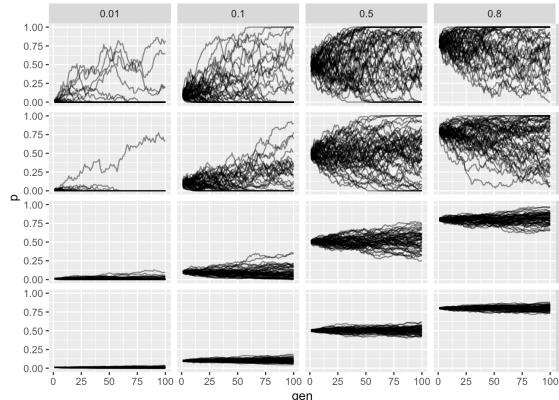
- much more amenable to biological realism

Disadvantages:

- less theory to back up behavior
- Wright-Fisher model has its own caveats
- historically very slow (until now...)

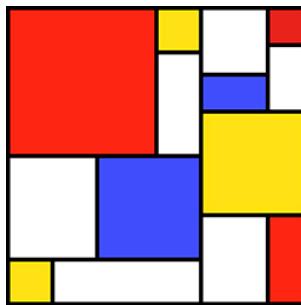
Examples:

- simuPOP (<http://simupop.sourceforge.net/>)
- SFS_CODE (http://sfscode.sourceforge.net/SFS_CODE/)
- SLiM (<https://messerlab.org/slim/>)



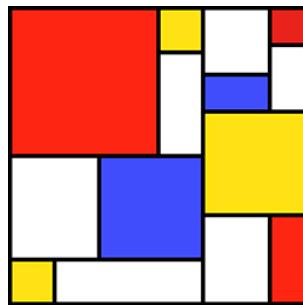
(<https://stephens999.github.io/>)

SLiM 3.0 and its advantages



- actually very fast
- tree sequence recording for even more speed and keeping track of local genealogy
 - scriptable*: Eidos language like R or python
- non Wright-Fisher module allows model violations like overlapping generations
 - integration with tskit and msprime through python
 - Nice RStudio-like IDE

SLiM 3.0 and its advantages



Haller, B.C., Messer, P.W. 2019. SLiM 3: Forward genetic simulations beyond the Wright–Fisher model. *Molecular Biology and Evolution* 36: 632–637. doi: <https://doi.org/10.1093/molbev/msy228>

Haller, B.C., Galloway, J., Kelleher, J., Messer, P.W., Ralph, P.L. 2019. Tree-sequence recording in SLiM opens new horizons for forward-time simulation of whole genomes. *Molecular Ecology Resources* 19: 552–566. doi: <https://doi.org/10.1111/1755-0998.12968>

Haller, B.C., Messer, P.W. 2019. Evolutionary modeling in SLiM 3 for beginners. *Molecular Biology and Evolution* 36: 1101–1109. doi: <https://doi.org/10.1093/molbev/msy237>

Tutorial: getting started

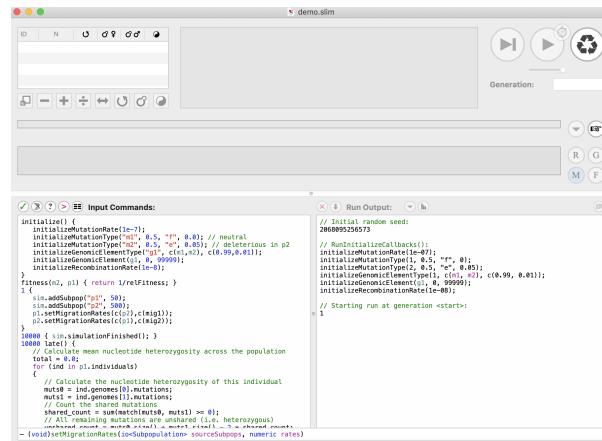
- 1) Download SLiM 3.3 from <https://messerlab.org/slim/> (Mac / Linux only)
- 2) Open Terminal and set your directory somewhere harmless:

```
set wd ~/Desktop/
```

- 2) Clone SliM tutorial repository:

```
git clone https://github.com/elinck/SLiM_tutorial
```

- 3) Open `hill_robertson.slim` in SliM GUI



Tutorial: SLiM gui

The screenshot shows the SLiM graphical user interface with several key components:

- Top Bar:** Includes standard OS X window controls (red, yellow, green buttons) and a title bar labeled "demo.slim".
- Left Panel:** A sidebar titled "Subpopulation information" containing a table with columns ID, N, ♂, ♀, ♂♀, and ♀♂. Below the table are various icons for file operations.
- Central Area:** Two main visualization panels:
 - A large panel titled "Visualization of individuals" which is currently empty.
 - A smaller panel titled "Visualization of chromosome(s)" which is also currently empty.
- Right Panel:** A control panel with buttons for "advance 1 gen", "run simulation", and "refresh". It also includes a "Generation:" input field and some circular icons for R, G, M, and F.
- Bottom Left:** A "Input Commands:" section containing the following SLiM code:

```
initialize() {  
    initializeMutationRate(1e-7);  
    initializeMutationType("m1", 0.5, "f", 0.0); // neutral  
    initializeMutationType("m2", 0.5, "e", 0.05); // deleterious in p2  
    initializeGenomicElementType("g1", c(m1,m2), c(0.99,0.01));  
    initializeGenomicElement(g1, 0, 99999);  
    initializeRecombinationRate(1e-8);  
}  
fitness(m2, p1) { return 1/relFitness; }  
1 {  
    sim.addSubpop("p1", 50);  
    sim.addSubpop("p2", 500);  
    p1.setMigrationRates(c(p2),c(mig1));  
    p2.setMigrationRates(c(p1),c(mig2));  
}  
10000 { sim.simulationFinished(); }  
10000 late() {  
    // Calculate mean nucleotide heterozygosity across the population  
    total = 0.0;  
    for (ind in p1.individuals)  
    {  
        // Calculate the nucleotide heterozygosity of this individual  
        muts0 = ind.genomes[0].mutations;  
        muts1 = ind.genomes[1].mutations;  
        // Count the shared mutations  
        shared_count = sum(match(muts0, muts1) >= 0);  
        // All remaining mutations are unshared (i.e. heterozygous)  
        unshared_count = muts0.size() + muts1.size() - 2 * shared_count;  
    }  
}
```
- Bottom Center:** A "Console" section showing the command-line output of the simulation setup:

```
// Initial random seed:  
2068095256573  
  
// RunInitializeCallbacks():  
initializeMutationRate(1e-07);  
initializeMutationType(1, 0.5, "f", 0);  
initializeMutationType(2, 0.5, "e", 0.05);  
initializeGenomicElementType(1, c(m1, m2), c(0.99, 0.01));  
initializeGenomicElement(g1, 0, 99999);  
initializeRecombinationRate(1e-08);  
  
// Starting run at generation <start>:  
1
```
- Bottom Right:** An "Output" section showing the initial random seed and the start of the simulation run.

Tutorial: Question 1

Do independent beneficial mutations affect each other's fixation probabilities?

Hill, W.G., Robertson, A. 1966. The effect of linkage on limits to artificial selection. *Genetics Research* 8: 269-294. doi:
<https://doi.org/10.1017/s001667230800949x>

Tutorial: Question 1

1) Set up simulation conditions:

```
initialize() {
    initializeMutationRate(1e-6); // mutation rate
    initializeMutationType("m1", 0.5, "f", 0.05); // mut. type
    initializeGenomicElementType("g1", m1, 1.0); // architecture
    initializeGenomicElement(g1, 0, 99999); // element length
    initializeRecombinationRate(1e-8); // recomb. rate
}
```

2) Add a population:

```
1 {
    sim.addSubpop("p1", 1000); // 1000 individuals
}
```

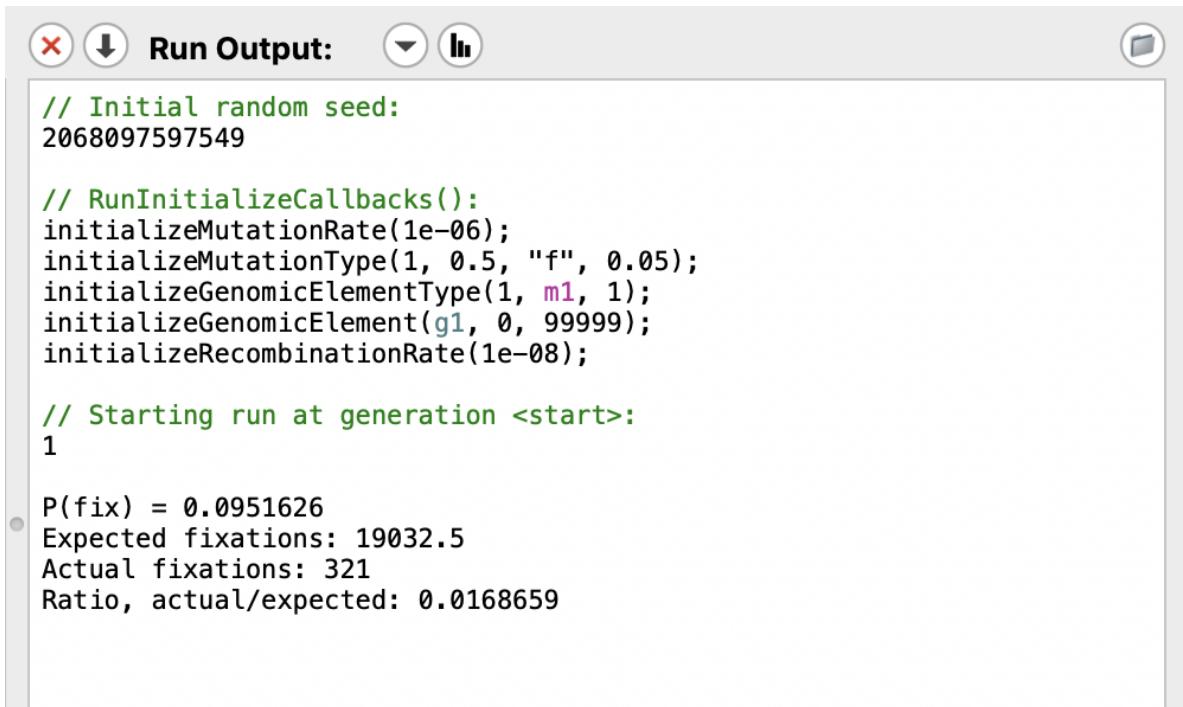
Tutorial: Question 1

3) End simulation and calculate things:

```
6000 late() {
// Calculate the fixation probability for a beneficial mutation
    s = 0.05;
    N = 1000;
    p_fix = (1 - exp(-2 * s)) / (1 - exp(-4 * N * s));
// Calculate the expected number of fixed mutations
    n_gens = 1000; // first 5000 generations were burn-in
    mu = 1e-6;
    locus_size = 100000;
    expected = mu * locus_size * n_gens * 2 * N * p_fix;
// Figure out the actual number of fixations after burn-in
    subs = sim.substitutions;
    actual = sum(subs.fixationGeneration >= 5000);

// Print a summary of our findings
cat("P(fix) = " + p_fix + "\n");
cat("Expected fixations: " + expected + "\n");
cat("Actual fixations: " + actual + "\n");
cat("Ratio, actual/expected: " + (actual/expected) + "\n");
}
```

Do independent beneficial mutations affect each other's fixation probabilities?



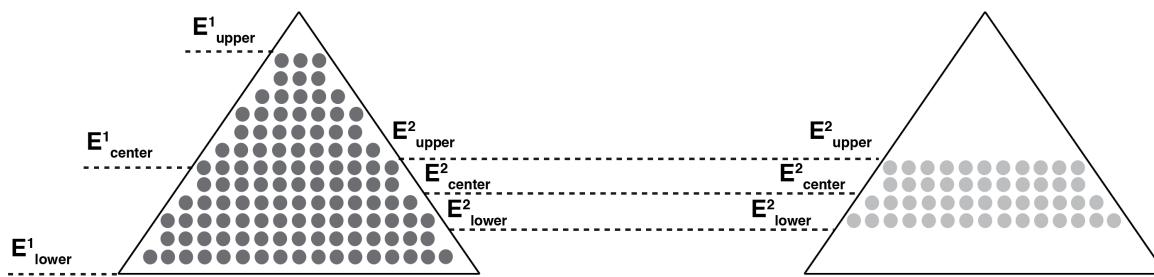
The screenshot shows a software interface with a title bar "Run Output:" and various control buttons. The main area displays a text log of a simulation run:

```
// Initial random seed:  
2068097597549  
  
// RunInitializeCallbacks():  
initializeMutationRate(1e-06);  
initializeMutationType(1, 0.5, "f", 0.05);  
initializeGenomicElementType(1, m1, 1);  
initializeGenomicElement(g1, 0, 99999);  
initializeRecombinationRate(1e-08);  
  
// Starting run at generation <start>:  
1  
  
P(fix) = 0.0951626  
Expected fixations: 19032.5  
Actual fixations: 321  
Ratio, actual/expected: 0.0168659
```

They do, which means recombination might be very useful!

Tutorial: Question 2

What are the consequences of an influx of maladaptive alleles for genetic diversity?



Tutorial: Question 2

1) Open `migration_selection.slim` in SliM GUI

2) Note deleterious mutation

```
initialize() {
    initializeMutationRate(1e-7);
    initializeMutationType("m1", 0.5, "f", 0.0); // neutral
    initializeMutationType("m2", 0.5, "e", 0.05); // deleterious i
    initializeGenomicElementType("g1", c(m1,m2), c(0.99,0.01));
    initializeGenomicElement(g1, 0, 99999);
    initializeRecombinationRate(1e-8);
}
```

3) Note fitness depends on population

```
fitness(m2, p1) { return 1/relFitness; }
```

Tutorial: Question 2

- 4) Note subpopulation sizes and that migration rate is a *variable*

```
1 {
    sim.addSubpop("p1", 50);
    sim.addSubpop("p2", 500);
    p1.setMigrationRates(c(p2),c(mig1));
    p2.setMigrationRates(c(p1),c(mig2));
}
```

Tutorial: Question 2

5) Note script to calculate pi, and output results:

```
10000 late() {
  total = 0.0;
  for (ind in p1.individuals)
  {
    muts0 = ind.genomes[0].mutations;
    muts1 = ind.genomes[1].mutations;
    shared_count = sum(match(muts0, muts1) >= 0);
    unshared_count = muts0.size() + muts1.size() - 2 * shared_
    pi_ind = unshared_count / (sim.chromosome.lastPosition + 1
    total = total + pi_ind;
  }
  pi = total / p1.individuals.size();
  line = paste(pi);
  cat("Mean nucleotide heterozygosity of p1 = " + pi + "\n");
  writeToFile("~/Dropbox/SLiM_tutorial/output.txt", line, append=T
}
```

Tutorial: Question 2

6) But to make this interesting, we need the results of multiple simulations. To do this, we'll run our script from the command line. Open terminal, and paste this bash loop, with the path adjusted as necessary:

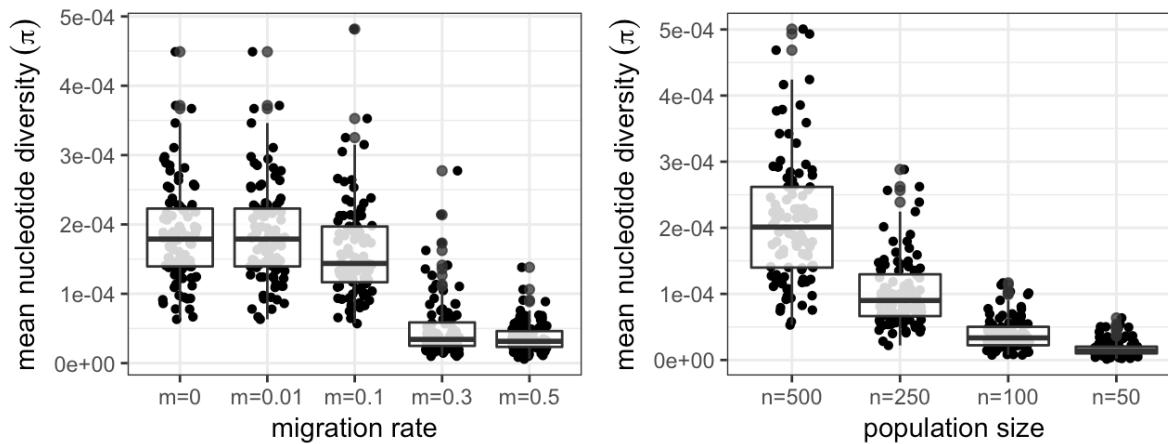
```
for rep in {1..100}
do
    slim -d mig1=0 -d mig2=0.1 migration_selection.slim
done
```

Note we set the migration variables here. Alternatively, simply run the file `loop.sh`.

```
bash loop.sh
```

- 7) Open up `output.txt` to see values of pi for each generation.
- 8) On your own time: run `migration_selection.slim` for m=0, m=0.01, m=0.1, and m=0.5. Visualize results with `plotting.R`.

What are the consequences of an influx of maladaptive alleles for genetic diversity?



With enough unidirectional migration, a strong reduction in effective population size

Resources

Online tutorial

https://github.com/elinck/SLiM_tutorial/blob/master/README.md

Molecular Ecologist post

<https://www.molecularecologist.com/2018/05/scriptable-evolutionary-simulations-in-slim-2/>

These slides

https://github.com/elinck/SLiM_tutorial/blob/master/SLiM_tutorial_slides.pdf

The massive, yet strangely readable manual

http://benhaller.com/slim/SLiM_Manual.pdf

Paper on SLiM for beginners

Haller, B.C., Messer, P.W. 2019. Evolutionary modeling in SLiM 3 for beginners. *Molecular Biology and Evolution* 36: 1101–1109. doi:

<https://doi.org/10.1093/molbev/msy237>

Google discussion group

<https://groups.google.com/forum/#!forum/slim-discuss>