# trying_out_pyslim.Rmd

## Wanting to see if ML trained on multiple coalescent slims can identify h and s

```
module load mamba
mamba create -n pyslim python pip
python -m pip install pyslim
```

```python
import pyslim
import tskit
import numpy as np
import msprime
import pandas
slim_tree = tskit.load("/u/scratch/e/eewade/swetha-sims123/sim_output_17_constant_demo_overlay.trees")

for recomb in c(1e-11, 1e-10, 1e-9, 1e-8, 1e-7, 1e-6):
  rts = pyslim.recapitate(slim_tree,
              recombination_rate=1e-8,
              ancestral_Ne=200, random_seed=5)

  # Verify it worked
  orig_max_roots = max(t.num_roots for t in slim_tree.trees())
  recap_max_roots = max(t.num_roots for t in rts.trees())
  print(f"Maximum number of roots before recapitation: {orig_max_roots}\nAfter recapitation: {recap_max_

  # msprime simulation
  new_time = 1000
  demog_model = msprime.Demography()
  demog_model.add_population(initial_size=10000, name='real')
  demog_model.add_population(initial_size=10000, name='fake')
  new_ts = msprime.sim_ancestry(
              samples={'real' : 10000, 'fake' : 1},
              demography=demog_model,
              end_time=new_time,
              sequence_length=rts.sequence_length,
              recombination_rate=1e-8,
              random_seed=9)

  H = new_ts.genotype_matrix()
  P = np.array([s.position for s in new_ts.sites()],dtype='float32')




# allele frequencies
p = slim_tree.sample_count_stat(
```

```
                    [slim_tree.samples()], lambda x: x/20000, 1, windows='sites',
                    span_normalise=False, polarised=True, strict=False)
print(f"There are {slim_tree.num_sites} segregating sites, of which {np.sum(p > 0.25)}")
print(f"are at frequency above 25%, and {np.sum(p > 0.05)} are above 5%.")
```