

```

1 import egglib
2
3 # specifies the list of loci to analyze
4 loci = [ 'data/align{}.fas'.format(str(i+1).rjust(4, '0'))
5                                                for i in range(100) ]
6
7 # initializes variables for computing averages
8 thetaW = 0.
9 D = 0.
10 nfixed = 0
11
12 # process all alignments
13 for locus in loci:
14
15     # imports the alignment
16     align = egglib.Align(locus)
17
18     # performs diversity analysis
19     pol = align.polymorphism()
20
21     # extracts statistics
22     print 'locus: {}'.format(locus)
23     print '    number of samples: {}'.format(pol['nseff'])
24     print '    number of sites: {}'.format(pol['lseff'])
25     print '    number of polymorphic sites: {}'.format(pol['S'])
26     print '    Tajima\'s D: {}'.format(pol['D'])
27     thetaW += pol['thetaW']
28     if pol['S'] == 0:
29         nfixed += 1
30     else:
31         D += pol['D']
32
33 # computes and reports averages
34 thetaW /= len(loci)
35 D /= (len(loci)-nfixed)
36 print 'Average Watterson\'s theta: {}'.format(thetaW)
37 print 'Average D: {}'.format(D)
38 print 'Number of fixed loci: {}'.format(nfixed)
39
40 # performs neutral coalescent simulations
41 theta = 1.
42 ns = 15
43 nrep = 1000
44 params = egglib.simul.CoalesceParamSet(ns)
45 mut = egglib.simul.CoalesceFiniteAlleleMutator(theta)
46 simuls = egglib.simul.coalesce(params, mut, nrep)
47
48 # computes average D over simulations
49 Dsim = 0
50 n = 0
51 for simul in simuls:
52     pol = simul.polymorphism()
53     if pol['S'] != 0:
54         Dsim += pol['D']
55         n += 1
56 Dsim /= n
57 print 'Average D from simulations: {}'.format(Dsim)
58

```