

```

%Code for simulating Constitutive Mutator
function mutator()
hold off
fixcount = 0;
for a=1:100 %the number of iterations can be changed
a
normal = [0.5,0,0,0,0,0,0,0,0,0];
mutator = [0.5,0,0,0,0,0,0,0,0,0];
mul = 1*10^(-5);
mud = 10^(-4);
mudr = 10^(-8);
mub = 10^(-8);
m = 100;
mutationnormal = [(1-mul-mud-mub) mud 0 0 mub 0 0 0 mul; mudr (1-mudr-mud-
mub-mul) mud 0 0 mub 0 0 mul;
0 mudr (1-mud-mudr-mul-mub) mud 0 0 mub 0 mul;
0 0 mudr (1-mudr-mub-mul) 0 0 0 mub mul;
0 0 0 0 (1-mud-mul) mud 0 0 mul;
0 0 0 0 mudr (1-mudr-mud-mul) mud 0 mul;
0 0 0 0 0 mudr (1-mud-mudr-mul) mud mul;
0 0 0 0 0 0 mudr (1-mudr-mul) mul;
0 0 0 0 0 0 0 0];

mmud = m*mud;
mmub = m*mub;
mmudr = m*mudr;
mmul = m*mul;
mutationmutator = [(1-mmud-mmud-mmub) mmud 0 0 mmub 0 0 0 mmul;
mmudr (1-mmudr-mmud-mmub-mmul) mmud 0 0 mmub 0 0 mmul;
0 mmudr (1-mmud-mmudr-mmul-mmub) mmud 0 0 mmub 0 mmul;
0 0 mmudr 1-mmudr-mmub-mmul 0 0 0 mmub mmul;
0 0 0 0 1-mmud-mmul mmud 0 0 mmul;
0 0 0 0 mmudr 1-mmudr-mmud-mmul mmud 0 mmul;
0 0 0 0 0 mmudr 1-mmud-mmudr-mmul mmud mmul;
0 0 0 0 0 0 mmudr 1-mmudr-mmul mmul;
0 0 0 0 0 0 0 0];

%For SIM mutator, we can switch to this mutation matrix
simmutationmutator = [(1-mmud-mmud-mmub) mmud 0 0 mmub 0 0 0 mmul;
mmudr (1-mmudr-mmud-mmub-mmul) mmud 0 0 mmub 0 0 mmul;
0 mmudr (1-mmud-mmudr-mmul-mmub) mmud 0 0 mmub 0 mmul;
0 0 mmudr 1-mmudr-mmub-mmul 0 0 0 mmub mmul;
0 0 0 0 1-mud-mul mud 0 0 mul;
0 0 0 0 mudr 1-mudr-mud-mul mud 0 mul;
0 0 0 0 0 mudr 1-mud-mudr-mul mud mul;
0 0 0 0 0 0 mudr 1-mudr-mul mul;
0 0 0 0 0 0 0 0];

b = 1.5;
d = 0.05;
fitness= [1,1-d,1-2*d,1-3*d,1+b,1+b-d,1+b-2*d,1+b-3*d,-1];
mutatorfreq = [];
normfreq = [];
benfreq=[];
for i = 1:1500
nf = fitness.*normal;
nm = fitness.*mutator;
total1 = sum(nf);
total2 = sum(nm);
newtotal = 1+total1+total2;

```

```

normal = (nf+normal)/newtotal;
mutator = (nm+mutator)/newtotal;
%mutation step
normal = normal*mutionnormal;
mutator = mutator*mutionmutator; %simply swap out with
simmutationmutator to simulate SIM
%random sampling by poisson
numnormal = 10^(6)*normal;
nummutator = 10^(6)*mutator;
samplernormal = [0 0 0 0 0 0 0 0 0];
samplermutator = [0 0 0 0 0 0 0 0 0];
%nummutator(1);
for j=1:9
    samplernormal(j) = poissrnd(numnormal(j));
    samplermutator(j) = poissrnd(nummutator(j));
end

samplesum = sum(samplernormal)+sum(samplermutator);
normal = samplernormal/samplesum;
mutator = samplermutator/samplesum;
mutatorfreq(i)=sum(mutator);
end
if(mutatorfreq(1500)>=0.99)
    fixcount=fixcount+1;
end
plot(mutatorfreq);
hold on;
end
fixcount

```