1. A sequence of population matrices from a Markov chain

2. Calculate stochastic growth rates from this sequence

Orou Gaoue | ogaoue@gmail.com

Sept 26, 2010; University of Miami, ITME

I used part of S. Ellner 2008

PART I: A function to get the sequence of states *********

inputs: iteration "nt" and Markov chain "A"

output: sequence of states "1", "2", "3" given the number of states of your Markov chain

```
sequences<-function(nt,A){
B = apply(A,2,cumsum); # cumulative sums of each column
A; B;

states=numeric(nt+1); # A vector of "nt" 0 to receive the sequence
rd=runif(nt);                 # Random uniform distr of norm "nt"
states[1] = 3;                # Start in open state

for(i in 1:nt) {
        b=B[,states[i]]; #cumulative probabilities for current state
        states[i+1]=sum(rd[i]>b)+1 # based on current state
        }
return(states) ## The function will yield a seq of states
}
```

Applying the function

```
nt = 100000;  ## The length of the sequence
A = matrix(c(0.98,0.10,0,0.02,0.7,0.05,0,0.2,0.95),3,3,byrow=T);

seq<-sequences(nt,A) ## This is the sequence of nt states

## plot(seq[1:1000],type="s"); ## ploting the sequence

## PART II: To get the sequence of matrix according to this sequence of environment "states"

## I made up 3 enviromnents A1, A2, A3 with pop matrices
A1<-matrix(c(0.98,0.10,0,0.2), 2,2, byrow=T) ## state 1
A2<-matrix(c(0.8,0.30,0,0.02), 2,2, byrow=T) ## state 2
A3<-matrix(c(0.6,0.10,0,0.4), 2,2, byrow=T)  ## state 3

nstates<-ncol(A)            ## number of environments stages
nstages<-ncol(A1)    ## number of life stages

## I create an array 'mat' of the 3 matrices
mat<-array(c(A1,A2,A3), c(nstages,nstages,nstates))

## I create an array 'matseq' to receive the nt seq of matrices
matseq<-array(0, c(nstages,nstages,nt))

for (i in 1:nt){
        matseq[,,i]<-mat[,,seq[i]]
        }
```

That gives a seq of nt matrix chosen between A1,A2,A3 and you can use that for your stochastic simulations.

PART III: Calculate the stochastic population g.r

```
egA1<-eigen(A1)

meig<-which(Re(egA1$values)==max(Re(egA1$values)))

W<-egA1$vectors

w<-abs(Re(W[,meig])) ## SSD: will be used as initial vectors n0


n0<-w                        ## Initial vector n0 = stable stage vector

tr<-2000              ## The transient phase to remove from simulation

r<-numeric(nt)  ## A vector to receive the "nt" g.r.
```

SIMULATION TO GET STOCH r: I use the matrix population model to calculate the stage distribution at each time step and the total population size. Then I calculate r=N(t+1)/N(t) for each time step, to get a sequence of one-time step pop growth rate r which will be average to get the stochastic long run population growth rate

```
for (t in tr:nt){
        n0<-matseq[,,t]%*%n0
        N<-sum(n0)
        r[t]<-log(N)
        n0<-n0/N
  }
```

The stochastic growth rate

```
stoch.gr<-exp(mean(r[tr:(nt-tr)]))  ## Removed the transient phase

SE<-1.96*sqrt(var(r[tr:(nt-tr)])/(nt-tr)) ## Standard Error

CI<-c(stoch.gr-SE, stoch.gr+SE)
```