**CBS810 Infectious Disease Modeling Homework**

**Simulation of Stochastic Model using Gillespie’s Algorithm**

Instructions:

* This assignment is due on Monday 7th December
* Please submit your work by email ([schen31@ncsu.edu](mailto:clanzas@utk.edu)) and be sure to include your scripts.

Suppose we are modeling an endemic system with ***disease-induced death***:

1. List the events, and associated transitions and rates of the stochastic model; the infection and recovery are already provided

|  |  |  |
| --- | --- | --- |
| **Event** | **Transition** | **Associated Rate** |
| **Infection** | **(S,I)→(S-1, I+1)** |  |
| **Recovery** | **(S,I)→(S+1, I-1)** |  |
| **Birth** | **(S)→(S+1)** |  |
| **Death** | **(I)→(I-1)** |  |

1. Complete a script to simulate the stochastic model.

# The Gillespie function

gillesp <- function(start,ratefun,trans,pars,times) {

t0 <- times[1] ## set time to starting time

ntimes <- length(times) ## total time duration

X <- start ## set state to starting state

res <- matrix(nrow=length(times),ncol=length(start),dimnames=list(times,names(start)))

## matrix for results

for (ctr in 1:(ntimes-1)) { ## loop over reporting times

res[ctr,] <- X ## record current state

while (t0<times[ctr+1]) { ## using while instead of for

rates <- ratefun(X,pars,t0) ## calculate current rates

if (all(rates==0)) break ## extinction

totrate <- sum(rates)

elapsed <- rexp(1,totrate) ## sample elapsed time

which.trans <- sample(1:nrow(trans),size=1,prob=rates) ## pick transition

t0 <- t0+elapsed ## update time

X <- X+trans[which.trans,] ## add transition values to current state

} }

cbind(times,res)

}

## starting condition

start=c(S=99,I=1, R=0)

# specify rate function: beta\*S\*I and gamma\*I for infection/transmission and recovery

ratefun.SIR = function(X,pars,time) {

vals = c(as.list(pars),as.list(X)) ## attach state and pars as lists

rates = with(vals, ##allows reference to states and parameters by name

c(infection=beta\*S\*I, recovery=gamma\*I, birth=mu\*S, death=mu\*I)) # specify the associated rates

}

statenames.SIR = c(“S”, “I”, “R” ) ## state variable names

transnames.SIR = c(“infection”, “recovery”, “birth”, “death”) # transition names

# transmission matrix

trans.SIR = matrix(c(-1, 1, 0, 0, -1, 1, 1, 0, 0, 0, -1, 0), # transition matrix

byrow=TRUE, ## default is by column

ncol=3, ## number of columns = number of state variables

dimnames=list(transnames.SIR,statenames.SIR))

# specify parameters

pars.SIR= c(beta=0.05, gamma=1, mu=0.2) # You can try different values

# specify simulation time

times= seq(0, 5, by = 0.05)

# run the simulation

G.SIR.mult= replicate(100, gillesp(start = start, times = times, ratefun = ratefun.SIR, trans = trans.SIR, pars = pars.SIR)[, "I"])

matplot(times,G.SIR.mult,type="l",col="gray",lty=1, xlab="Time", ylab="Number of Infectious")

lines(times,rowMeans(G.SIR.mult),lwd=2)