

Deterministic model solution code:

```
library("deSolve")
library("tikzDevice")

## Vector Field for SIR Metapatch Model
SIRmeta.vector.field <- function(t,vars,parms=NULL) {
  ##parms should be of form (R_0,gamma,mu,alpha,m,EC)
  ##vars should be of form (S,I,R)
  with(as.list(c(parms,vars)), {
    n <- length(vars)/3
    S <- vars[1:n]
    I <- vars[(n+1):(2*n)]
    R <- vars[(2*n+1):(3*n)]

    #Equal Coupling
    if(EC==TRUE){
      M <- matrix(m,n,n)+(1-m)*diag(n) ##connectivity matrix
      betam <- R_0*(gamma+mu)*(1+alpha*cos(2*pi*t/365))*M ##beta matrix
    }else{
      #Nearest Neighbors
      M <- diag(n)
      M[row(M)%%n==(col(M)+1)%%n ] <- m
      M[row(M)%%n==(col(M)-1)%%n ] <- m
      betam <- R_0*(gamma+mu)*(1+alpha*cos(2*pi*t/365))*M ##beta matrix
    }

    dS <- NULL
    dI <- NULL
    dR <- NULL
    for (i in 1:n){
      dS[i] <- mu*(1-S[i]) - S[i]*sum(betam[i,]*I) ##dS_i/dt
      dI[i] <- S[i]*sum(betam[i,]*I) - (mu+gamma)*I[i] ## dI_i/dt
      dR[i] <- gamma*I[i] -mu*R[i]
    }
    vec.fld <- c(dS=dS,dI=dI,dR=dR)
    return(list(vec.fld))
  })
}

## Draw Solutions and coherence measure
draw.soln <- function(n,ic=c(x=rep(1,10),y=rep(0,10)),tmax=1,
  times=seq(0,tmax,
    by=tmax/1000),
  func,parms,...) {
  soln <- ode(ic,times,func,parms)

  #Plot I for all patches with different colours
  for (i in 1:n){
    lines(times/365,soln[,paste0('I',i)],col=i,...)
  }

  #Measure of coherence
  coherence <- lapply(1:length(times), function(tims) sqrt(sum((soln[tims,paste0('I',1:n)]-mean(soln[tims,1:n]))^2)/n))
}
```

```

lines(times/365,coherence,lwd=2,...)

labels <- c(1:n)
for(i in 1:n){
  labels[i]<-paste('$I_{',i,}'$',collapse=NULL)
}
legend("topright",legend=labels,col=1:n,lty=1,bty='n')
legend(x=22,y=0.0005, legend="$\\| I - \\langle I \\rangle e \\|$",bg="white")
}

#Parameters
R_0 <- 17 ##basic reproduction rate (if the system had no forcing)
gamma <- 1/13 ##inverse of mean infectious period
mu <- 1/(50*365) ##death and birth rate
alpha <- 0.1 ##strength of seasonal forcing
n <- 10 ## Number of patches
m <- 0.2 ##Connectivity matrix parameter
EC <- FALSE ##if true use equal coupling, if false use nearest neighbors

##Initial conditions
#S0 <- seq(0.3,0.7,(0.7-0.3)/(n-1))
S0 <- seq(0.04,0.1, (0.1-0.04)/(n-1))
I0 <- rep(1e-4,n)
R0 <- 1-S0-I0

tmax<-10000

tikz("detNNR017m0.2.tex",standAlone=TRUE,width=6,height=6)
## draw box for plot
plot(0,0,xlim=c(0,tmax/365),ylim=c(0,0.006),type="n",xlab="Time (years)",
     ylab="Prevalence (I)",las=1)

## draw solutions
draw.soln(n,ic=c(S=S0,I=I0,R=R0),tmax=tmax,
          func=SIRmeta.vector.field,
          parms=c(R_0,gamma,mu,alpha,m,EC))
dev.off()

## tikz output
##      2

tools::texi2dvi('detNNR017m0.2.tex',pdf=T)

```

Stochastic model simulations code:

```

set.seed(899)
SIRmeta.Gillespie <- function(tmax,ic=c(S0,I0,R0),parms=c(R_0=2,gamma=0.25,mu=4e-5,alpha=0.1,m=0.2,EC=T)
  times <- 0

  n <- length(ic)/3
  #S, I, R matrices: column index is patch index and rows (added later in while loop)
  #are values at different time steps
  S <- matrix(0,1,n)
  I <- matrix(0,1,n)

```

```

R <- matrix(0,1,n)
#Initial Coniditions
S[1,] <- ic[1:n]
I[1,] <- ic[(n+1):(2*n)]
R[1,] <- ic[(2*n+1):(3*n)]

index <- 1
while(times[index] < tmax){
  #A matrix of all rates
  rates <- matrix(0,n,6)
  #the first column is all birth rates
  rates[,1] <- mu*(S[index,]+I[index,]+R[index,])
  #thrid column are recovery rates
  rates[,3] <- gamma*I[index,]
  #last three columns are death rates from S and I
  rates[,4] <- mu*S[index,]
  rates[,5] <- mu*I[index,]
  rates[,6] <- mu*R[index,]
  #Second column are transmission rates
  #Equal Coupling
  if(EC==TRUE){
    M <- matrix(m,n,n)+(1-m)*diag(n) ##connectivity matrix
    betam <- R_0*(gamma+mu)*(1+alpha*cos(2*pi*times[index]/365))*(1/(1+(n-1)*m))*M ##beta matrix
  }else{
    #Nearest Neighbors
    M <- diag(n)
    M[row(M)%%n==(col(M)+1)%%n ] <- m
    M[row(M)%%n==(col(M)-1)%%n] <- m
    betam <- R_0*(gamma+mu)*(1+alpha*cos(2*pi*times[index]/365))*(1/(1+2*m))*M ##beta matrix
  }
  rates[,2] <- S[index,]*colSums(t(betam[1:n,])*I[index,])
  totalrate <- sum(rates)

  #Getting a timestep using R's exp distribution
  timestep <- rexp(n=1, rate= totalrate)
  times <- rbind(times, times[index]+timestep)

  #Adding new row to S, I, R
  S <- rbind(S, S[index,])
  I <- rbind(I, I[index,])
  R <- rbind(R, R[index,])

  #Getting which compartment transistion occurred
  #TO DO: Do a bisection search (or some other faster method) instead
  #Get a random number from a uniform distribution
  randomvar <- runif(n=1,min=0,max=totalrate)
  interval <- 1
  #Make the rates matrix into a vector for convinence
  rateslisted <- c(0,as.vector(t(rates)))
  #Find which interval randomval is in from list: [0,a_1), [a_1, a_1+a_2), etc
  #If in interval [a_1 + .. + a_(i-1), a_1 + ... + a_i), transition given by rate a_i occurred
  while(interval < (6*n)){
    if(randomvar >= sum(rateslisted[1:interval]) & randomvar < sum(rateslisted[1:(interval+1)])){

```

```

        break
      }else{
        interval<- interval+1
      }
    }
    #Get row, col indices of a_i in rates matrix
    #row index says in which patch a transition occurred
    #col index says what type of transition occurred
    colInd <- interval%%6
    rowInd <- ceiling(interval/6)
    if(colInd ==1){ ##A birth occurred
      S[index+1,rowInd] <- S[index,rowInd]+1
    }else if(colInd ==2){ ##An infection occurred
      S[index+1,rowInd] <- S[index,rowInd]-1
      I[index+1,rowInd] <- I[index,rowInd]+1
    }else if(colInd ==3){ ##A recovery occurred
      I[index+1,rowInd] <- I[index,rowInd]-1
      R[index+1,rowInd] <- R[index,rowInd]+1
    }else if(colInd ==4){ ##A death in S occurred
      S[index+1,rowInd] <- S[index,rowInd]-1
    }else if(colInd ==5){ ##A death in I occurred
      I[index+1,rowInd] <- I[index,rowInd]-1
    }else if(colInd ==0){ ##A death in R occurred
      R[index+1,rowInd] <- R[index,rowInd]-1
    }

    index <- index+1
  }
  return(cbind(times, S, I, R))
}

tmax <- 1000
n <- 10
##Initial conditions
#S0 <- seq(45,90,(90-45)/(n-1))
#I0 <- rep(1,n)
#R0 <- 1000-S0-I0

S0 <- seq(225,450,(450-225)/(n-1))
I0 <- rep(3,n)
R0 <- 5000-S0-I0

#Parameters
R_0 <- 17 ##basic reproduction rate (if the system had no forcing)
gamma <- 1/13 ##inverse of mean infectious period
mu <- 1/(50*365) ##death and birth rate
alpha <- 0.1 ##strength of seasonal forcing
n <- 10 ## Number of patches
m <- 0.5 ##Connectivity matrix parameter
EC <- TRUE ##if true use equal coupling, if false use nearest neighbors

results <- SIRmeta.Gillespie(tmax,ic=c(S0,I0,R0),parms=c(R_0,gamma,mu,alpha,m,EC))
times<-results[,1]

```

```

I <- results[(n+1):(2*n+1)]

tikz("stochasticm0.5.tex",standAlone=TRUE,width=6,height=6)
## draw box for plot
plot(0,0,xlim=c(0,tmax/365),ylim=c(0,20),xlab="Time (years)",
     ylab="Prevalence (I)")
for (i in 1:n) {
  lines(times/365, I[,i],col=i) # use a different line colour for each patch
}
dev.off()

## tikz output
##          2

tools::texi2dvi('stochasticm0.5.tex',pdf=T)

```