Supplementary material of:
Combining food-web theory and population dynamics to assess the impact of invasive species

R-code part 2: Simulations of population dynamics

Chloé Vagnon, Rudolf P. Rohr

June 2021

Example: Simulations of population dynamics

Loading library and data

```r
library("deSolve") # Package for calculations

load("DATA2.Rdata") # Species inventory created in the 1st part of the code example
load("Mb.Rdata")    # Binary interaction matrix
load("Wji.Rdata")   # Species proportions in consumer diet

# Load the parameters to convert body size to metabolic rate
load("bodymass.Rdata") # For conversion from body size to bodymass
load("metabolic.Rdata") # For conversion from body mass to metabolic rate
```

A) Parametrization of the dynamic model with constants from literature and allometric relationships

```r
S <- length(DATA2$Species)      # Define the number of species (S)
Iprod <- which(colSums(Mb) == 0) # Define the primary producers (Iprod)
Cons <- which(colSums(Mb) > 0)   # Define the consumers (Cons)

# Log10(body size) is converted in Log10(body mass)
DATA2$Log10bm <- m_bodymass$coefficients[1] + m_bodymass$coefficients[2] * DATA2$Log10bs
# Log10(body mass) is converted in metabolic rate
DATA2$Log10mr <- m_metabolic$coefficients[1] +
  m_metabolic$coefficients[2]*DATA2$Log10bm + m_metabolic$coefficients[3]*DATA2$Log10bm^2

#Biotic capacity of primary producers
DATA2$K[Iprod] <- 10^(-0.77*DATA2$Log10bm[Iprod]-6)

# Maximum consumption rate
DATA2$y[DATA2$Category == 'Vertebrate'] <- 4
DATA2$y[DATA2$Category == 'Invertebrate' & DATA2$Diet=="invP"] <- 8
DATA2$y[DATA2$Category == 'Invertebrate' & DATA2$Diet=="inv"] <- 5
DATA2$y[DATA2$Category == 'Zooplankton'] <- 6.5
DATA2$y[DATA2$Diet == 'prod'] <- 1.69
```

```r
# Allometric constant(Brose et al., 2006)
DATA2$ax[DATA2$Category %in% 'Vertebrate'] <- 0.88
DATA2$ax[DATA2$Category != 'Vertebrate'] <- 0.314

# Efficiency of predator consumption
DATA2$epsilon[DATA2$Carnivorous == 1] <- 0.85
DATA2$epsilon[DATA2$Carnivorous ==0] <- 0.45

# Handling time
h<-t(Mb)
for (j in 1:nrow(h)){
  #for invertebrates
 if(is.element(rownames(h)[j],
              DATA2$Species[DATA2$Category%in%c("Invertebrate","Zooplankton")])){
   h[j,which(h[j,]!=0)]<-1/DATA2$y[j]
   }
 else{
 h[j,which(h[j,]!=0)]<-(4.084*10^5)*(10^DATA2$Log10bm[is.element(DATA2$Species,
                                          names(which(h[j,]!=0)))])*
   (10^(DATA2$Log10bm[j])^-0.75)
 }

}
```

B) Initialization of the simulations

The dynamic of primary producer i is given by:

$$\frac{dN_i}{dt} = r_i \cdot \left(1 - \frac{N_i}{K_i}\right) \cdot N_i - \sum_{j \in consumers\ of\ i} F_{ji}(\overrightarrow{N}) \cdot N_j \quad (eq.2)$$

Where the functional response (*i.e.*, the per capita consumption rate of consumer j on resource i) is given by:

$$F_{ji}(\overrightarrow{N}) = \frac{x_j \cdot y_j \cdot \omega_{ji} \cdot N_i}{1 + \sum_{k \in resources\ of\ j} h_{jk} \cdot x_j \cdot y_j \cdot \omega_{jk} \cdot N_k} \quad (eq.3)$$

The dynamic of consumer j is given by:

$$\frac{dN_j}{dt} = -m_i \cdot \left(\frac{N_j}{K_i}\right) \cdot N_i + \varepsilon_j \sum_{i \in resources\ of\ j} F_{ji}(\overrightarrow{N}) \cdot N_j - \sum_{j \in consumers\ of\ i} F_{lj}(\overrightarrow{N}) \cdot N_l \quad (eq.4)$$

```r
# Model used to calculate abundances at each time step
dN <- function(t,N,p){
  Fij <- p$alpha / as.vector(1 + (p$h *p$alpha)  %*%  N)
  out <- N * (p$r - p$alpha_intra * N  +  p$epsilon * Fij %*% N - t(Fij) %*% N)
  return(list(out))
}
# Note that alpha is the product of the metabolic rate mass dependant (xj), the maximum
# consumption rate (yj) and the matrix of resource proportions indiet of consumers (wij).
# It represents an interaction force, allometrically parameterized in our study but that
# can be parameterized with other technics.
```

C) Calulation of the abundances along time

```r
N_rand <- 10                    # Number of simulations
time_step <- seq(0,19999,1) # Time steps
Alive_sp <- rep(NA,N_rand)  # Vector allocation for alive species
Output<- array(NA,c(N_rand,length(time_step),S)) # Large array for stocking outputs


for (i in 1:N_rand){
  assign("last.warning", NULL, envir = baseenv())

  #Metabolic rate and noise simulated with the rnorm
  xj<- 10^(DATA2$Log10mr-DATA2$Log10bm)+ rnorm(S, mean=0, sd=0.001*(10^DATA2$Log10mr))


  # Growth
  r <- xj
  r[Cons] <- -r[Cons] * DATA2$ax[Cons]


  # alpha
  alpha <- (xj * DATA2$y) * t(Wji)
  alpha_intra <- rep(0,nrow(alpha))
  alpha_intra[Iprod] <- r[Iprod]/DATA2$K[Iprod] #intraspecific regulation for Iprod

  #List of parameters to induce in the equa diff computing
  p <- list(alpha = alpha, alpha_intra = alpha_intra, r = r, epsilon = DATA2$epsilon, h=h)

  # Initialization of initial  abundances
  N0 <- sort(round(runif(S, min=0.15, max=1), digits=5), decreasing = F)

  # Calculatin of results from differential equations
  out <- ode(y = N0, times = time_step, func = dN, parms = p)

  if (length(warnings())==0){    #Avoid warning messages
    N_equ <- out[dim(out)[1],-1] # N at the equilibrium
    alive <- N_equ > 1e-6 # Computing of species still alive

    Alive_sp[i] <- sum(alive)
    Output[i,,] <- as.matrix(out[,-1])
  }
}
```