Equations & R Code

Seasonal (within-year) dynamics are modeled as three differential equations:

$$\frac{\mathrm{d}D_i}{\mathrm{d}\tau} = 0 \tag{1}$$

$$\frac{\mathrm{d}N_i}{\mathrm{d}\tau} = N_i \epsilon_i f_i(R) \tag{2}$$

$$\frac{\mathrm{d}R}{\mathrm{d}\tau} = -\sum_{i=1,2} f_i(R) N_i \tag{3}$$

```
## Continuous model
updateDNR <- function(t, DNR, parms){
  with(as.list(c(DNR, parms)), {
    dD1dt = 0
    dD2dt = 0
    dN1dt = N1*eps[1]*(uptake_R(r[1], R, alpha[1], beta[1]))
    dN2dt = N2*eps[2]*(uptake_R(r[2], R, alpha[2], beta[2]))
    dRdt = -1 * (dN1dt + dN2dt)
    list(c(dD1dt, dD2dt, dN1dt, dN2dt, dRdt)) #output
  })
}

##  Resource uptake function (Hill function)
uptake_R <- function(r, R, alpha, beta){
  return((r*R^alpha) / (beta^alpha + R^alpha))
}
```

where i denotes species, D is the dormant (long-lived) biomass state, and N is the living biomass (fast-growing, shorter-lived) state, and ϵ_i is each species' resource-to-biomass conversion efficiency. The growth rate of living biomass is a resource-dependent Hill function, $f_i(R) = r_i R^{a_i}/(b_i^{a_i} + R^{a_i})$, where r is a species' intrinsic growth rate and a and b define the curvature of the function. Resource depletion is equal to the sum of each species' consumption, $\sum_{i=1,2} f_i(R)N_i$.

At the beginning of each season we start with initial conditions defined as V_t, W_t, and Z_t for the dormant state, the live state, and the resource, respectively. So for each season, eqs. 1-3 are solved given the initial conditions:

$$D_i(0) = V_{i,t} \tag{4}$$

$$N_i(0) = W_{i,t} \tag{5}$$

$$R(0) = Z_t \tag{6}$$

The consumers transition between N and D instantaneously between years. So, at the yearly transition:

$$V_{i,t+1} = \alpha_i N_i(T^-) + D_i(T^-)(1-\gamma_{i,t})(1-\eta_i) \tag{7}$$

$$W_{i,t+1} = \beta_i(1-\alpha_i)N_i(T^-) + \gamma_{i,t}[D_i(T^-) + \alpha_i N_i(T^-)](1-\eta_i) \tag{8}$$

$$Z_{t+1} = \theta_i(1-\alpha_i)N_i(T^-) + \nu R(T^-) + R(T^-) \tag{9}$$

```
## Discrete model
update_DNR <- function(t,DNR,gs,a1,a2,m1,m2,b1,b2,theta1,theta2,c){
  with (as.list(DNR),{
    g1 <- gs[1]
    g2 <- gs[2]
    D1 <- a1*N1 + D1*(1-g1)*(1-m1)
    D2 <- a2*N2 + D2*(1-g2)*(1-m2)
    N1 <- b1*(1-a1)*N1 + g1*(D1+(a1*N1))*(1-m1)
    N2 <- b2*(1-a2)*N2 + g2*(D2+(a2*N2))*(1-m2)
    R <- theta1*(1-a1)*N1 + theta2*(1-a2)*N2 + c*R + Rvector[t]
    return(c(D1, D2, N1, N2, R))
  })
}
```

where $D(T^{-})$, $N(T^{-})$, and $R(T^{-})$ are the densities of each state at the end of the year and g is a time-fluctuating activation rate that regulates how much dormant biomass is converted to growing-season live biomass each year. $R(T^{+})$ is a randomly generated resource pulse from a log-normal distribution with mean R_{μ} and variance R_{σ^2}. Parameter notations and definitions are in table 1.

Table 1: Definition of model states and parameters. ‘Constant’ parameters do not fluctuate during a simulation, while ‘variable’ parameters do fluctuate through time in a given simulation.

Parameter	Definition
N	live biomass
D	dormant biomass
R	resource supply
α	allocation fraction of live biomass to seed bank (constant)
β	live state survivorship (constant)
η	dormant state mortality rate (constant)
ϵ	resource-to-biomass efficiency (constant)
γ	dormant-to-live biomass transition fraction (variable)
ν	resource carry-over fraction (constant)
θ	resource recycling fraction (constant)
a	Hill function power parameter (constant)
b	Hill function parameter (constant)
ρ	correlation of species’ responses transition cue (constant)
σ_E^2	variance of transition cue (constant)
R_μ	mean resource pulse, on log scale (constant)
R_{σ^2}	variance of annual resource supply, on log scale (variable)

Full R Code for Simulations

```
####
#### Initial Conditions, Global Variables, and Parameters ----------------------
####
seasons <- 1000                  # number of seasons to simulate
seasons_to_exclude <- 200        # initial seasons to exclude from plots
days_to_track <- 20              # number of days to simulate in odSolve
DNR <- c(D=c(1,1),N=c(1,1),R=10) # initial conditions
Rmu <- 3                         # mean resource pulse (on log scale)
Rsd_annual <- 0                  # std dev of resource pulses (on log scale)
sigE <- 2                        # environmental cue variance
rho <- 0                         # environmental cue correlation between species

# Within-season parameters
parms <- list(
  r = c(5,5),                    # max growth rate for each species
  alpha = c(5,5),                # rate parameter for Hill function
  beta = c(20,20),               # shape parameter for Hill function
  eps = c(0.2,0.2)               # resource-to-biomass efficiency
)

# End-of-season transition parameters
a1 <- 0.50                       # live-to-dormant biomass fraction; spp1
a2 <- 0.50                       # live-to-dormant biomass fraction; spp2
b1 <- 0                          # adult survivorship; spp1 (0 if annual, >0 if perennial)
b2 <- 0                          # adult survivorship; spp2 (0 if annual, >0 if perennial)
m1 <- 0.1                        # dormant mortality; spp1
m2 <- 0.1                        # dormant mortality; spp2
```

```
theta1 <- 0                      # resource recycling fraction; spp1
theta2 <- 0                      # resource recycling fraction; spp2
c <- 0                           # resource carry-over fraction

####
#### Load relevant libraries ----------------------------------------
####
library('deSolve')
library('mvtnorm')



####
#### Model functions -------------------------------------------------
####
## Continuous model
updateDNR <- function(t, DNR, parms){
  with(as.list(c(DNR, parms)), {
    dD1dt = 0
    dD2dt = 0
    dN1dt = N1*eps[1]*(uptake_R(r[1], R, alpha[1], beta[1]))
    dN2dt = N2*eps[2]*(uptake_R(r[2], R, alpha[2], beta[2]))
    dRdt = -1 * (dN1dt + dN2dt)
    list(c(dD1dt, dD2dt, dN1dt, dN2dt, dRdt)) #output
  })
}

##  Resource uptake function (Hill function)
uptake_R <- function(r, R, alpha, beta){
  return((r*R^alpha) / (beta^alpha + R^alpha))
}

## Discrete model
update_DNR <- function(t,DNR,gs,a1,a2,m1,m2,b1,b2,theta1,theta2,c){
  with (as.list(DNR),{
    g1 <- gs[1]
    g2 <- gs[2]
    D1 <- a1*N1 + D1*(1-g1)*(1-m1)
    D2 <- a2*N2 + D2*(1-g2)*(1-m2)
    N1 <- b1*(1-a1)*N1 + g1*(D1+(a1*N1))*(1-m1)
    N2 <- b2*(1-a2)*N2 + g2*(D2+(a2*N2))*(1-m2)
    R <- theta1*(1-a1)*N1 + theta2*(1-a2)*N2 + c*R + Rvector[t]
    return(c(D1, D2, N1, N2, R))
  })
}


####
#### Simulate model -----------------------------------------------------
####
days <- c(1:days_to_track)

# Get "germination" fractions for each year
```

```
getG <- function(sigE, rho, nTime){
  varcov <- matrix(c(sigE, rho*sigE, rho*sigE, sigE), 2, 2)
  e <- rmvnorm(n = nTime, mean = c(0,0), sigma = varcov)
  g <- exp(e) / (1+exp(e))
  return(g)
}

# Loop over seasons
nms <- names(DNR)
gVec <- getG(sigE = sigE, rho = rho, nTime = seasons)
Rvector <- rlnorm(seasons, Rmu, Rsd_annual)
saved_outs <- matrix(ncol=5, nrow=seasons)
for(season_now in 1:seasons){
  output <- ode(y = DNR, times=days,
                func = updateDNR, parms = parms)
  DNR <- output[nrow(output),nms]
  saved_outs[season_now,] <- DNR
  names(DNR) <- nms
  DNR <- update_DNR(season_now, DNR, gVec[season_now,],
                    a1=a1,a2=a2,m1=m1,m2=m2,b1=b1,b2=b2,
                    theta1=theta1,theta2=theta2,c=c)
  names(DNR) <- nms
}
```