

# Bears and Salmon: a Food Web Model of a Lake ecosystem

Exercises Accompanying the Course Environmental Modelling

Karline Soetaert and Dries Bonte, Ghent University

July 2021

## Questions

### Problem formulation

In the previous lecture you made a food-web model of salmon, bears, scavengers and salmon carcasses. You first created the conceptual model (flow diagram), then wrote the mass balance equations.

Now, you will create a rate expression for every flow that entered the mass balance equations. Once you have done that, you can see how this model is implemented in R.

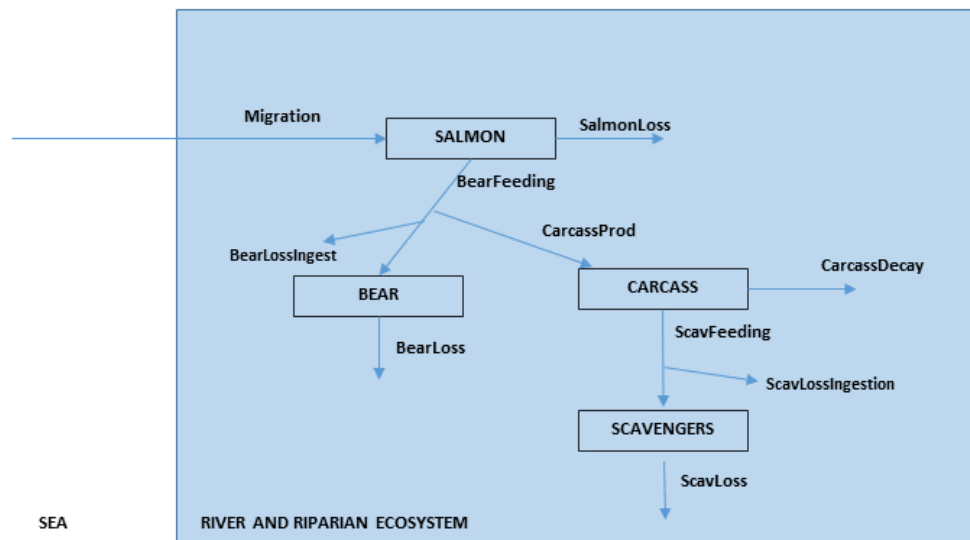


Figure 1: Bears and Salmon

### Mass balance equations

Suitable mass balances for the state variable SALMON, BEAR, CARCASS and SCAVENGERS are:

$$\frac{dSALMON}{dt} = Migration - BearFeeding - SalmonLoss$$

$$\begin{aligned}\frac{dBEAR}{dt} &= BearFeeding - CarcassProd - BearLossIngest - BearLoss \\ \frac{dCARCASS}{dt} &= CarcassProd - ScavFeeding - CarcassDecay \\ \frac{dSCAVENGERS}{dt} &= ScavFeeding - ScavLossIngest - ScavLoss\end{aligned}$$

## Assumptions

The assumptions are the following:

1. The units of the state variables are  $kg\ C\ m^{-2}$ ; the time unit is days.
  - Question: What are the units of the flows?
2. The model domain (river + surrounding area) is  $1000\ m \times 1000\ m$  large.
3. The input of the salmon in the river is defined by a parameter called *migrate*. Per day, 10000 *kg* of salmon carbon migrate up the river.
4. The mortality + spawning + basal respiration of salmon, the hunting and basal respiration of the bear, and the losses of the scavengers are expressed by a first-order rate.
5. The parameters that you should use, and their values are:

parameter	units	description
area = 1000000	$[m^2]$	
migrate = 10000	$[kg\ C\ d^{-1}]$	Amount of salmon entering the river system
rSalmonLoss = 0.05	$[d^{-1}]$	Salmon loss rate constant (Spawning+death+basalResp)
rBearFeeding = 0.02	$[d^{-1}]$	Bear feeding rate constant
ksSalmon = 0.01	$[kg\ C\ m^{-2}]$	Half saturation ct for ingestion bear
rBearLoss = 0.01/365	$[d^{-1}]$	Death rate constant (hunting and other) for bear
pLossToCarcass = 0.5	$[-]$	Losses of salmon to carcass (fraction)
pBearLossIngest = 0.4	$[-]$	Ingested fraction that is lost (respiration+faeces)
rCarcassDecay = 0.001	$[d^{-1}]$	Decay rate constant of carcasses
rScavFeeding = 0.02	$[d^{-1}]$	Feeding rate constant of scavengers
ksCarcass = 0.002	$[kg\ C\ m^{-2}]$	Half saturation ct for ingestion scavenger
pScavLossIngest = 0.7	$[-]$	Growth efficiency of scavengers
rScavLoss = 1/365	$[d^{-1}]$	Mortality+basal respiration rate constant scavengers

## Tasks

The above information is all you get. Write the mathematical equations for each flow on a piece of paper.

Note: to make the equations (and the model code) easy to follow, we use the following formalism:

- State variables are written in CAPITAL LETTERS,
- Flows start with a capital letter, and
- parameters start with a lower case letter.

If you have finished with this exercise, then you can try to read, understand and then run the implementation of this model in R, using R markdown.

It is in the file called *RiverRun\_Rcode.Rmd*. The model is run twice, with different rate constants characterising ingestion by the bears.

- Try to explain the output of this model.

- Assuming the mean weight of a bear at the start of the simulation is 300 *kg*, how many bears are there in the area? What is their final weight?

## Answers

The equations to create are relatively simple.

The only “complication” is the specification of *Migration*; the parameter specifies the total mass in kg C entering the area per day; we need to divide by the area ( $m^2$ ) to arrive at a change in  $kg\ C\ m^{-2}\ d^{-1}$ .

Feeding by the bear and by the scavengers are regular ecological interactions (*BearFeeding*, *ScavFeeding*), where the bear or the scavengers are the active compartment. Thus the grazers (BEAR and SCAVENGER) set the maximal rate, while the food (SALMON, CARCASS) determines the rate limiting term, modelled by a Monod function.

The model is carbon-based. Carbon is used both as a *substrate* for the production of new biomass, but also to deliver *energy*, via respiration. There are two respiration terms:

- the basal respiration, which maintains the organism’s basal metabolism. This is represented as a fixed fraction of the biomass per day. For simplicity, this term is merged here with the other loss rates of the organisms (e.g., due to mortality) (rates *SalmonLoss*, *BearLoss*, *ScavengerLoss*).
- activity respiration, which delivers the energy that is required for the production of new biomass from the food. This is represented as a fixed fraction of the assimilated food (*assimilation* = *ingestion* – *faecesproduction*).

$$BearFeeding = rBearFeeding \times \frac{SALMON}{(SALMON + ksSalmon)} \times BEAR$$

$$Migration = migrate/area$$

$$SalmonLoss = rSalmonLoss \times SALMON$$

$$CarcassProd = pLossToCarcass \times BearFeeding$$

$$BearLoss = rBearLoss \times BEAR$$

$$BearLossIngest = pBearLossIngest \times (BearFeeding - CarcassProd)$$

$$ScavFeeding = rScavFeeding \times \frac{CARCASS}{(CARCASS + ksCarcass)} \times SCAVENGER$$

$$ScavLossIngest = pScavLossIngest \times ScavFeeding$$

$$ScavLoss = rScavLoss \times SCAVENGER$$

$$CarcassDecay = rCarcassDecay \times CARCASS$$

## Model implementation in R

Here is the R-code specifying the model:

```
require(deSolve) # package with solution methods

# state variables, units = kgC/m2
state <- c(BEAR = 0.01, SALMON = 0, SCAVENGER = 0.005, CARCASS = 0.001)
```

```

# parameters
parms <- c(
  area          = 1000000, # [m2]
  migrate       = 10000,   # [kg C/day] Amount of salmon entering the river system
  rSalmonLoss   = 0.05,    # [/day] Spawning, death, respiration or salmon rate constant

  rBearFeeding  = 0.02,    # [/day] Bear feeding rate constant
  ksSalmon      = 0.01,    # [kg C/m2] Half saturation coeff for feeding of bear
  rBearLoss     = 0.01/365, # [/day] Death rate constant (hunting and other losses)
  pLossToCarcass = 0.5,    # [-] Loss fraction of killed salmon to carcass
  pBearLossIngest = 0.4,   # [-] Part lost after ingestion (activity resp + faeces production)

  rCarcassDecay = 0.001,   # [/day] Decay rate constant of carcasses

  rScavFeeding  = 0.02,    # [/day] Scavenger feeding rate constant
  ksCarcass     = 0.002,   # [kg C/m2] Half saturation coeff for feeding of scavenger
  pScavLossIngest = 0.7,   # [-] Part lost after ingestion (activity resp + faeces production)
  rScavLoss     = 1/365    # [/day] Scavenger Loss rate constant (mortality and respiration)
)

# Model function
RiverRun <- function(t, state, params) {
  with (as.list(c(state, params)), {

    # Rate expressions - all in units of [kgC/m2/day]

    # Salmon dynamics
    Migration      <- migrate/area
    SalmonLoss     <- rSalmonLoss * SALMON

    # Bear dynamics
    BearFeeding    <- rBearFeeding * SALMON / (SALMON + ksSalmon) * BEAR
    CarcassProd    <- pLossToCarcass * BearFeeding
    BearLossIngest <- pBearLossIngest * (BearFeeding - CarcassProd)
    BearLoss       <- rBearLoss * BEAR

    # Scavenger dynamics
    ScavFeeding    <- rScavFeeding * CARCASS / (CARCASS + ksCarcass) * SCAVENGER
    ScavLossIngest <- pScavLossIngest * ScavFeeding
    ScavLoss       <- rScavLoss * SCAVENGER

    # Carcasses
    CarcassDecay   <- rCarcassDecay * CARCASS

    # Mass balances [gC/m2/day]
    dBEAR         <- BearFeeding - CarcassProd - BearLossIngest - BearLoss
    dSALMON        <- Migration - BearFeeding - SalmonLoss
    dCARCASS       <- CarcassProd - CarcassDecay - ScavFeeding
    dSCAVENGER     <- ScavFeeding - ScavLossIngest - ScavLoss
    list(c(dBEAR, dSALMON, dSCAVENGER, dCARCASS))
  })
}

```

The model is solved twice with different parameter values, and the output plotted.

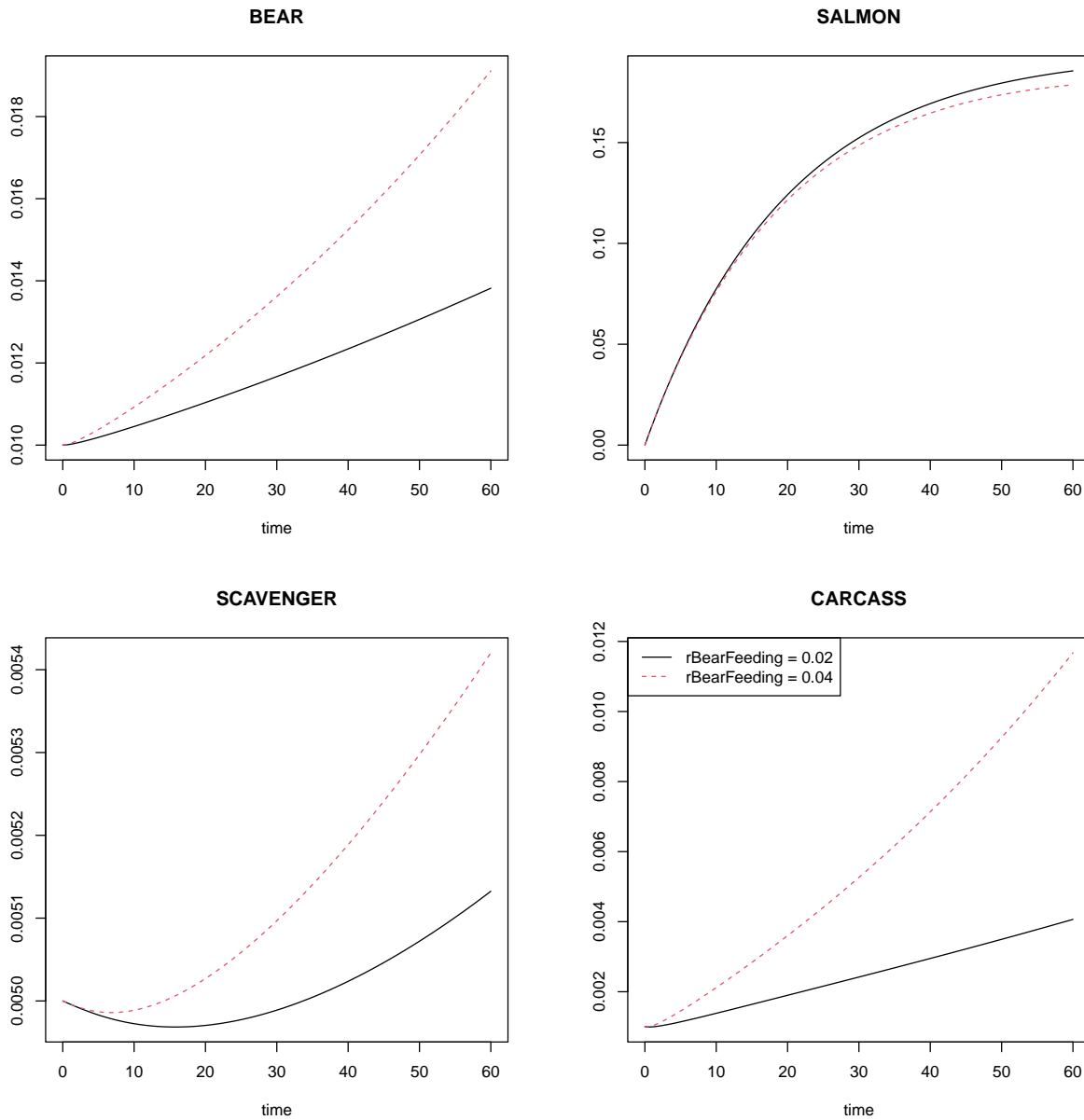
```
# output time
outtimes <- seq(from = 0, to = 60, length.out = 100) # run for 2 months

# solve this model using the ode function from package deSolve
out <- ode(y = state, parms = parms, func = RiverRun, times = outtimes)

# Change the parameter values: bears are now more efficient to capture salmon
parms2 <- parms
parms2["rBearFeeding"] <- 0.04 # instead of 0.02
out2 <- ode(y = state, parms = parms2, func = RiverRun, times = outtimes)

# visualise it (black = out, red = out2)
plot(out, out2)

legend("topleft", c("rBearFeeding = 0.02", "rBearFeeding = 0.04"), col = 1:2, lty = 1:2)
```



## References

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Karline Soetaert, Thomas Petzoldt, R. Woodrow Setzer (2010). Solving Differential Equations in R: Package deSolve. Journal of Statistical Software, 33(9), 1–25. URL <http://www.jstatsoft.org/v33/i09/> DOI 10.18637/jss.v033.i09