What is Lotka-Volterra Model?

The Lotka–Volterra model is frequently used to describe the dynamics of ecological systems in which two species, the predator (y) and the prey (x), interacting each other in limited space. The Lotka–Volterra model assumes that the prey consumption rate by a predator is directly proportional to the prey abundance. This means that predator feeding is limited only by the amount of prey in the environment.

In Figure 1, I designed a conceptual model with *Noctiluca sp.* as a predator and *Cymbella sp.* as a prey. *Noctiluca* is a marine dinoflagellate, which is large and has an increase population at night. *Noctiluca* feeds on various minute marine organisms, including also marine diatoms. *Cymbella sp.* on the other hand is a marine diatom, which has a high population growth rate, as a primary producents of photosynthesis.

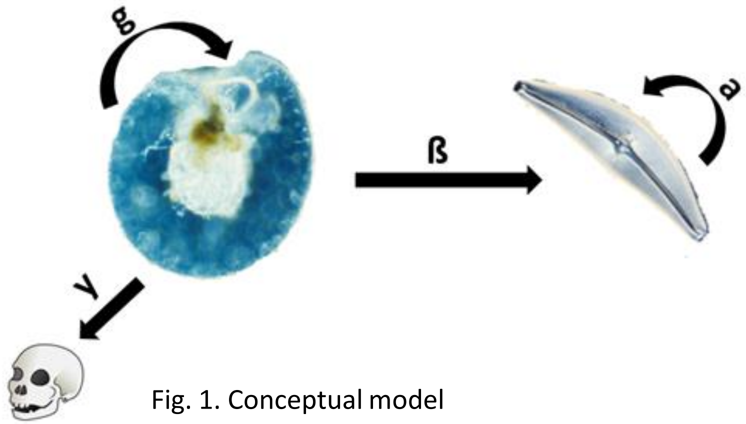

Fig. 1. Conceptual model

On this table are given the parameters, which we need to calculate the Lotka-Volterra Model, in this case the prey-predator behavior between *Noctiluca sp.* and *Cymbella sp.*

Parameter	
a	Prey population growth rate
β	Predation rate
δ	Predator population growth rate
γ	Predator mortality rate
x	Prey species
y	Predator species

The equations governing the dynamics of the prey and predator are: The prey grows at a linear rate (alpha) and gets eaten by the predator at the rate of (beta). The predator gains a certain amount vitality by eating the prey at a rate (delta), while dying off at another rate (gamma).

$$\frac{dx}{dt} = ax - \beta xy$$

$$\frac{d\gamma}{dt} = \delta xy - \gamma y$$

In Figure 2 we see the behavior of a prey (a) and a predator (b) when they are not interacting with each other. They have both a high growth rate, when there are not other factors, which impact their growth rate.

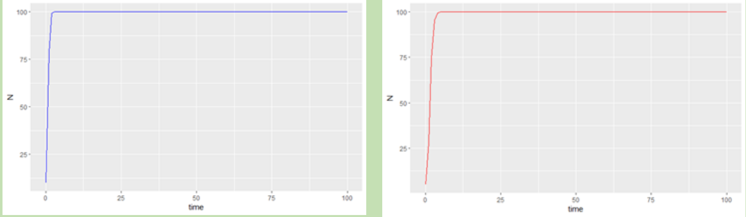

Fig 2. a) Prey b) Predator

When they are competing with each other, we have an other outcome.

Results

This choice of parameters leads to periodic dynamics in which the prey population initially increases, leading to an abundance of food for the predators. The predators increase in response (lagging the prey population), eventually overwhelming the prey population, which crashes. This in turn causes the predators to crash, and the cycle repeats. The period of these dynamics is about 15 seconds, with the predators lagging the prey by about a second.

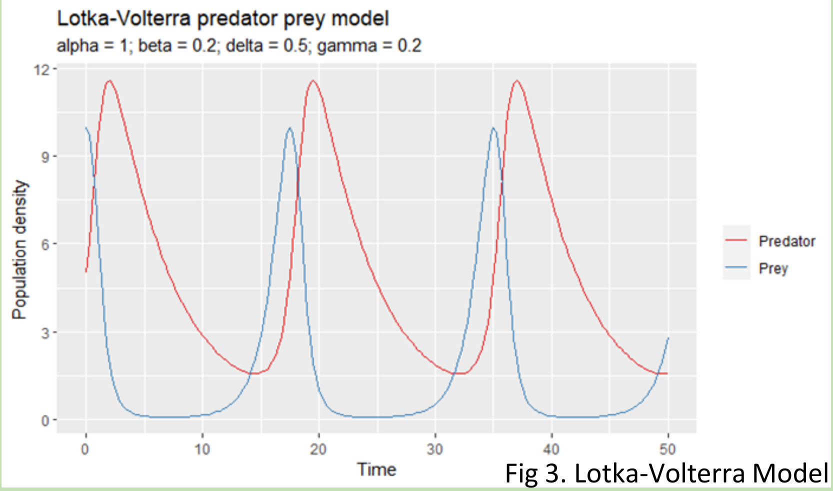

Fig 3. Lotka-Volterra Model

Equilibrium

The system is at equilibrium if the variables don‘t change anymore and are static. In a continuous-time model, the derivatives over time are 0. In Fig 3 there is no equilibrium because the variables are not static. But here you can find stable cycles, because the height and the bottom part of the model-curve are repeating again and again in cycles.

I used the R packages “deSolve” to solve a system of differential equations and “ggplot2” to create the plot.

```
## Lotka-Volterra Model ##
library(tidyverse)
library(deSolve)
pars <- c(alpha = 1, beta = 0.2, delta = 0.5, gamma = 0.2)
init <- c(x = 10, y = 5)
times <- seq(0, 100, by = 1)
deriv <- function(t, state, pars) {
  with(as.list(c(state, pars)), {
    d_x <- alpha * x - beta * x * y
    d_y <- delta * beta * x * y - gamma * y
    return(list(c(x = d_x, y = d_y)))
  })
}
lv_results <- ode(init, times, deriv, pars)
lv_model <- function(pars, times = seq(0, 50, by = 1)) {
  state <- c(x = 10, y = 5)
  deriv <- function(t, state, pars) {
    with(as.list(c(state, pars)), {
      d_x <- alpha * x - beta * x * y
      d_y <- delta * beta * x * y - gamma * y
return(list(c(x = d_x, y = d_y)))
})
 }
  ode(y = state, times = times, func = deriv, parms = pars)
lv_results <- lv_model(pars = pars, times = seq(0, 50, by = 0.25))
lv_results %>%
data.frame() %>%
gather(var, pop, -time) %>%
mutate(var = if_else(var == "x", "Prey", "Predator")) %>%
ggplot(aes(x = time, y = pop)) +
  geom_line(aes(color = var)) +
  scale_color_brewer(NULL, palette = "Set1") +
  labs(title = "Lotka-Volterra predator prey model",
  subtitle = paste(names(pars), pars, sep = " = ", collapse = "; "),  x = "Time", y = "Population density")
```