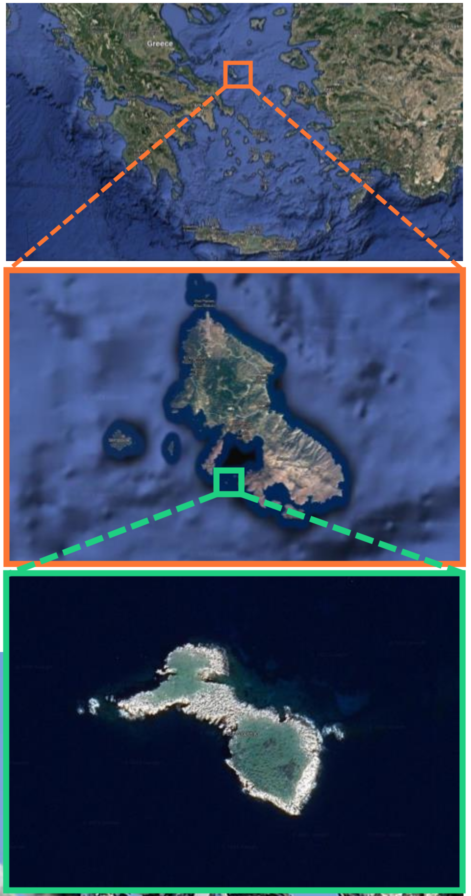
Group: Amalia Bogri, Christian Berrig & Jonas Bolduan

Competition systems

General model parameters

Parameter	Calculation	Unit
Beetles, N		$[\#N/m^2]$
Longevity, λ		$[\text{days}]$
Death rate	$\mathrm{d} = 1/\lambda$	$[1/\text{days}]$
Offspring, φ		$[1/\text{year}]$
Feeding frequency, f_{fr}		$[meal/day]$
Feeding quantity, f_q		$[g/meal]$
Feeding per year, f_{yr}	$f_{yr} = 365 \cdot f_{fr} \cdot f_q$	$[g/year]$
Max consumption, c_{max}		$[g/\#N \cdot day]$
Efficiency, ε	$\varepsilon = \varphi/f_{yr}$	$[\#N/g]$
Clearance rate, b		$[m^2/\#N \cdot day]$
Guano, R		$[g/m^2]$
Influx-outflux rate, r		$[1/day]$
Carrying capacity, K		$[g/m^2]$

LOCATION:
Exo Diavates,
$V = 20\ [m^2]$



General model description:

RESOURCES i – Guano: R

$$\frac{dR_i}{dt} = r_i \cdot (K_i - R_i) - \sum_j C(i,j)N_j$$

CONSUMERS j – Beetles: N

$$\frac{dN_j}{dt} = C(i,j)N_j - d_j \cdot N_j$$

CONSUMPTION:
Functional response for consumer j

$$C(i,j) = \frac{\rho_j \cdot c_{max}}{\rho_j + c_{max}}$$

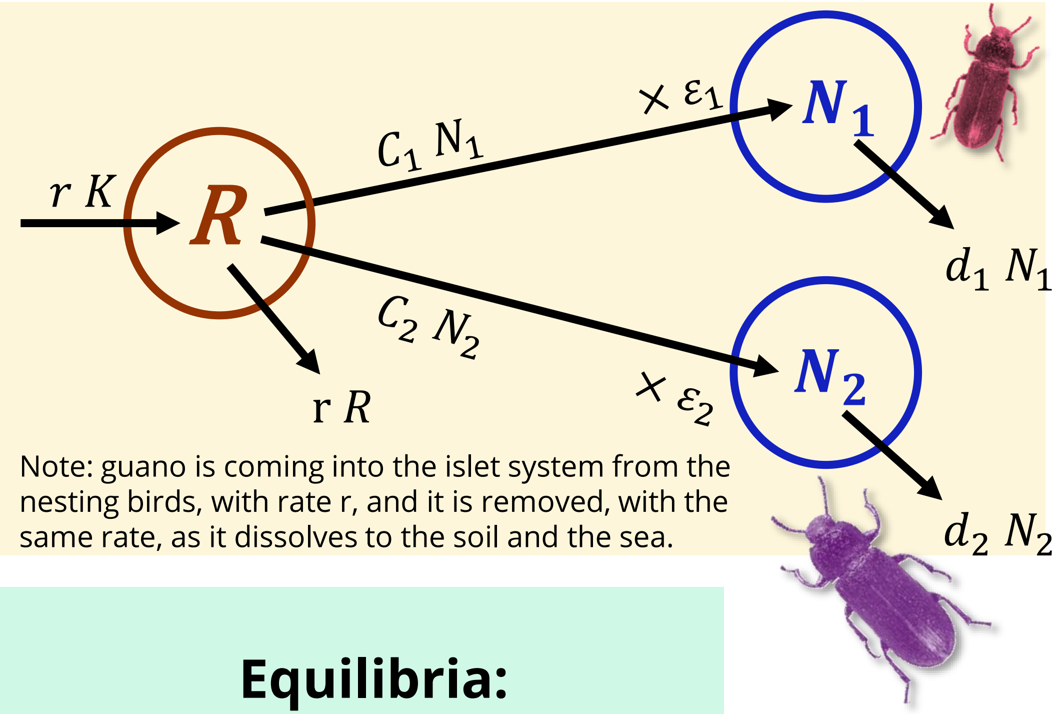
For SUBSTITUTIONAL resources:

$$\rho_j = \sum_i R_i \cdot b_{ij}$$

For ESSENTIAL resources:

$$\rho_j = \min(R_j \cdot b_{ij})$$

Case 1: Two consumers & One resource



Note: guano is coming into the islet system from the nesting birds, with rate r, and it is removed, with the same rate, as it dissolves to the soil and the sea.

$$\frac{dR}{dt} = r \cdot (K - R) - \frac{b_1 \cdot R \cdot C_{max1} \cdot N_1}{b_1 \cdot R + C_{max1}} - \frac{b_2 \cdot R \cdot C_{max2} \cdot N_2}{b_1 \cdot R + C_{max2}} \quad \left[\frac{g}{m^2 \cdot day}\right]$$

$$\frac{dN_1}{dt} = N_1 \cdot \left(\varepsilon_1 \cdot \frac{b_1 \cdot R \cdot C_{max1}}{b_1 \cdot R + C_{max1}} - d_1\right) \quad \left[\frac{\#N1}{m^2 \cdot day}\right]$$

$$\frac{dN_2}{dt} = N_2 \cdot \underbrace{\left(\varepsilon_2 \cdot \frac{b_2 \cdot R \cdot C_{max2}}{b_2 \cdot R + C_{max2}} - d_2\right)} \quad \left[\frac{\#N2}{m^2 \cdot day}\right]$$

Equilibria:

$$R_j^* = \frac{d_i \cdot C_{maxj}}{\varepsilon_i \cdot C_{maxj} - d_i} \frac{1}{b_i}$$

$$N_j^* = \frac{r \cdot (K - R^*) \cdot \varepsilon_i}{d_i}$$

Initial growth rate:

$$G_i = \varepsilon_i \cdot \frac{b_i \cdot R_0 \cdot C_{maxi}}{b_i \cdot R_0 + C_{maxi}} - d_i$$

If R_0 is low, then the species with low death rate will have an advantage at the beginning.
If R_0 is high, then the species with higher efficiency, clearance rate and maximum consumption will have an advantage.

Case 1: Estimation of values

Parameter	Calculation	Unit	Values N1	Values N2	ValuesR
Beetles, $\boldsymbol{N}$		$[\#\boldsymbol{N}/\boldsymbol{m}^2]$			
Longevity, λ		[days]	365	365	-
Death rate	$\mathrm{d} = 1/\lambda$	$[1/days]$	1/365	1/365	-
Offspring, φ		$[1/year]$	4	4	-
Feeding frequency, f_{fr}		$[meal/day]$	1	1	-
Feeding quantity, f_q		$[g/meal]$	0.01	0.01	-
Feeding per year, f_{yr}	$f_{yr} = 365 \cdot f_{fr} \cdot f_q$	$[g/year]$	3.65	3.65	-
Max consumption, c_{max}		$[g/\#N \cdot day]$	0.10	0.12	-
Efficiency, ε	$\varepsilon = \varphi/f_{yr}$	$[\#N/g]$	1.09	1.09	-
Clearance rate, b		$[m^2/\#N \cdot day]$	0.0000012	0.0000012	-
Guano, $\boldsymbol{R}$		$[\boldsymbol{g}/\boldsymbol{m}^2]$			
Influx-outflux rate, r		$[1/day]$	-	-	10/365
Carrying capacity, K		$[g/m^2]$	-	-	10000

We do realize that b looks very small, but it is the only way to produce a sensible graph.

Case 1: Results

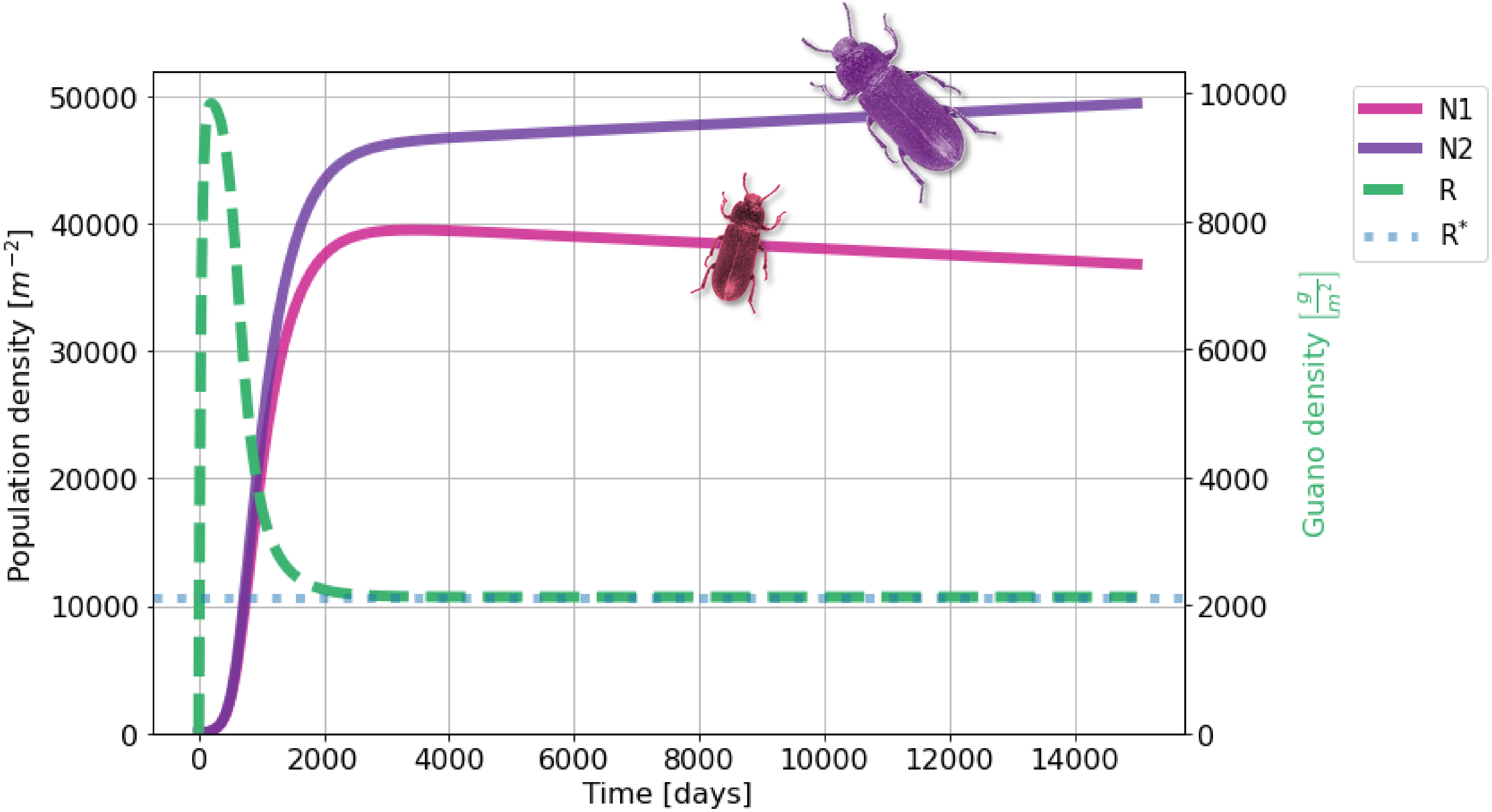
Note: N2 is also expected to ‘win’ at the start, as it has higher Cmax. However, since the system starts with 0 resources, the higher Cmax does not give an advantage at the first time step.

Initial parameters:
$R_0 = 0$
$N_{10} = 50$
$N_{20} = 50$

Equilibrium:
For N1: $R_1^* = 2136$
For N2: $R_2^* = 2127$

N2 has a lower R* and thus will win the competition (eventually).

The graph shows the transient phase, since the system is not stable for the amount of time the simulation was run for.

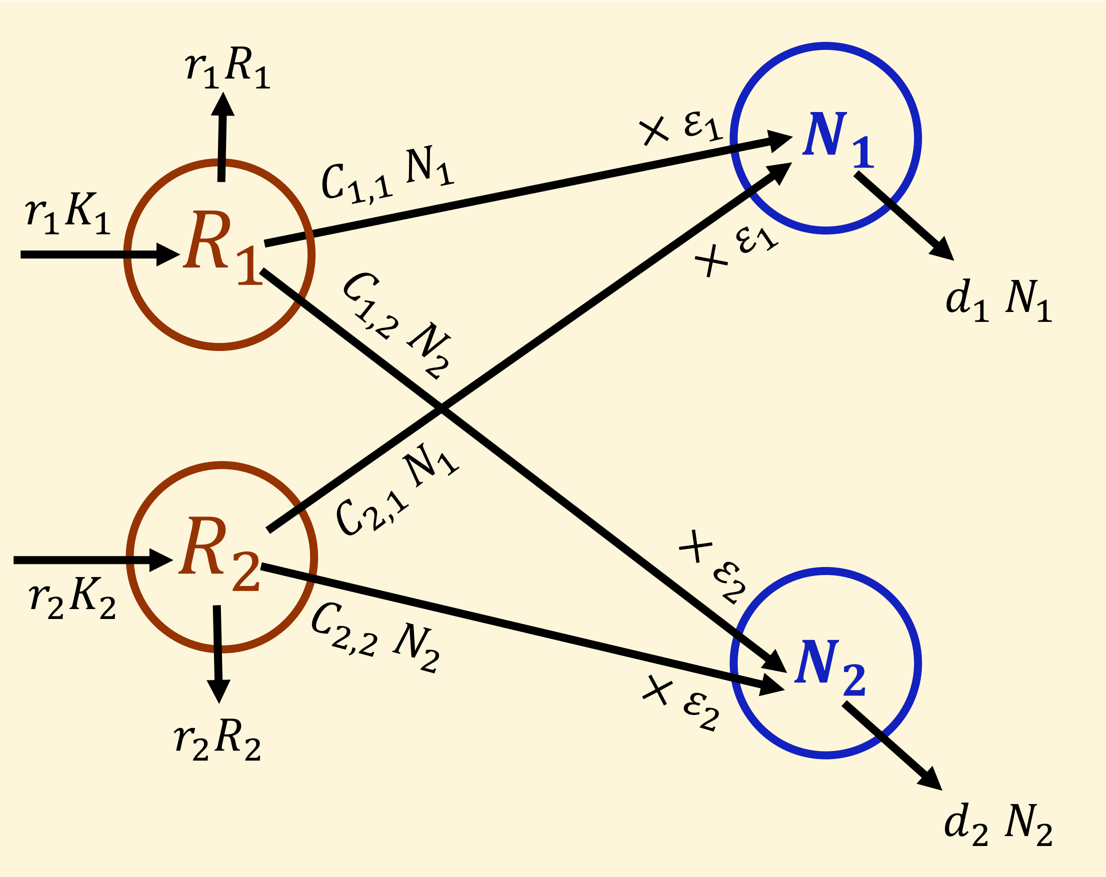


Case 1: Code in Python

```python
from scipy.integrate import odeint
import matplotlib.pyplot as plt
import numpy as np

# This is for reasonable fontsize universally defined:
fs_label = 16
parameters = {
                'figure.titlesize': fs_label+6,
                'axes.labelsize': fs_label,
                'axes.titlesize': fs_label+4,
                'xtick.labelsize': fs_label,
                'ytick.labelsize': fs_label,
                'legend.fontsize': fs_label,
                'lines.linewidth': 6
             }
plt.rcParams.update(parameters)


def deriv(state, t, *params):
    eps, bs, ds, c_max, r, K, nR = params
    Rs, Ns = state[:nR], state[nR:]
    R = Rs[0]
    C = lambda i: R*bs[i]*c_max[i]/(R*bs[i] + c_max[i])
    dR_dt = r*K - r*R - sum(C(i)*Ns[i] for i in range(len(Ns)))
    dN_dt = np.zeros(len(Ns))
    for iN, N in enumerate(Ns):
        dN_dt[iN] = eps[iN]*C(iN)*N - ds[iN]*N
    return np.array([dR_dt, *dN_dt])
```

```python
yr = 365 # day/yr
feed_freq = 1 # 1/day (eats every day)
feed_quant = 0.01 # g (single meal quant)
feed_yr = feed_quant*feed_freq*yr # g/yr
offspring = 4 # offspr/(beetle*yr)
eps = [offspring/feed_yr]*2 # 1/g  # Unit: [#N/g]
bs = [0.0000012, 0.0000012] # sqm /(day*beetle)
ds = [1/yr, 1/yr]  # Unit: [1/day]
c_maxs = [0.1, 0.12] # g/(beetle*day)
r = 10/yr  # Unit: [1/day]
K = 10000 # Unit: [g/sqm]
nR = 1 # number of distinct resources
V = 20 # Unit: [sqm]

init_state = [0, 50, 50]
t = np.linspace(0, 15000, 100000)
params = (eps, bs, ds, c_maxs, r, K, nR )
sol = odeint(deriv, init_state, t, params).T
R, N1, N2 = sol

print("Rstar:")
print(Rstar(0), Rstar(1))

fig, ax1 = plt.subplots(1,1, figsize=(12, 6), tight_layout=True)
ax2 = ax1.twinx()
ax1.grid()

ax2.plot(t, R, label="R", color='mediumseagreen', linestyle="--")
ax1.plot(t, N1, label="N1", color='mediumvioletred', alpha=0.8)
ax1.plot(t, N2, label="N2", color='rebeccapurple', alpha=0.8)
ax1.plot([], [], label="R", color='mediumseagreen', linestyle="--")

ax1.set_ylabel('Population density $\\left[m^{-2}\\right]$')
ax2.set_ylabel('Guano density $\\left[ \\frac{g}{m^{2}} \\right]$', color = 'mediumseagreen')
ax1.set_xlabel('Time [days]')
ax1.set_ylim(bottom=0)
ax2.set_ylim(bottom=0)
ax2.axhline(Rstar(1), linewidth=5, alpha=0.5, linestyle=":")
ax1.plot([], [], linewidth=5, alpha=0.5, linestyle=":", label="R$^{*}$")
ax1.legend(bbox_to_anchor=(1.15, 1), loc=2, borderaxespad=0.5, fontsize=15)
```

Case 2: Two consumers & Two substitutable resources



$$\frac{dR_1}{dt} = r_1 \cdot (K_1 - R_1) - \frac{b_{1,1} \cdot R \cdot C_{max1} \cdot N_1}{b_{1,1} \cdot R + C_{max1}} - \frac{b_{1,2} \cdot R \cdot C_{max2} \cdot N_2}{b_{1,2} \cdot R + C_{max2}} \quad \left[\frac{g}{m^2 \cdot day}\right]$$

$$\frac{dR_2}{dt} = r_2 \cdot (K_2 - R_2) - \frac{b_{1,1} \cdot R \cdot C_{max1} \cdot N_1}{b_{1,1} \cdot R + C_{max1}} - \frac{b_{1,2} \cdot R \cdot C_{max2} \cdot N_2}{b_{1,2} \cdot R + C_{max2}} \quad \left[\frac{g}{m^2 \cdot day}\right]$$

$$\frac{dN_1}{dt} = N_1 \cdot \left(\varepsilon_1 \cdot \frac{(b_1 \cdot R_1 + b_2 \cdot R_2) \cdot C_{max1}}{(b_1 \cdot R_1 + b_2 \cdot R_2) \cdot R + C_{max1}} - d_1\right) \quad \left[\frac{\#N1}{m^2 \cdot day}\right]$$

$$\frac{dN_2}{dt} = N_2 \cdot \left(\varepsilon_2 \cdot \frac{(b_1 \cdot R_1 + b_2 \cdot R_2) \cdot C_{max2}}{(b_1 \cdot R_1 + b_2 \cdot R_2) + C_{max2}} - d_2\right) \quad \left[\frac{\#N2}{m^2 \cdot day}\right]$$

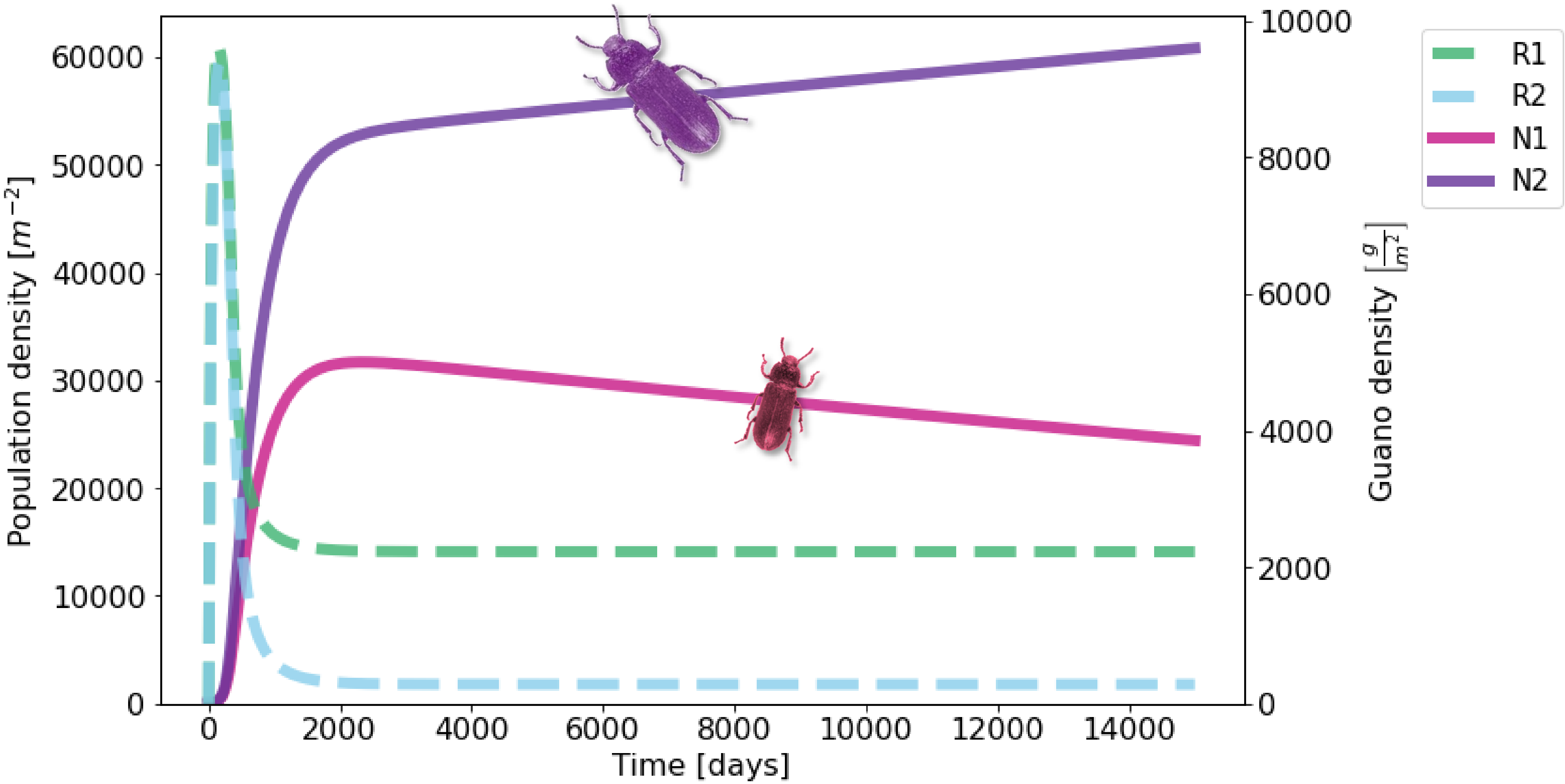
Case 2: Estimation of values


Parameter	Calculation	Unit	Values N1	Values N2	Values R1	ValuesR1
Beetles, $\boldsymbol{N}$		$[\#\boldsymbol{N}/\boldsymbol{m}^2]$				
Longevity, λ		[days]	365	365	-	-
Death rate	$\mathrm{d} = 1/\lambda$	$[1/days]$	1/365	1/365	-	-
Offspring, φ		$[1/year]$	4	4	-	-
Feeding frequency, f_{fr}		$[meal/day]$	1	1	-	-
Feeding quantity, f_q		$[g/meal]$	0.01	0.01	-	-
Feeding per year, f_{yr}	$f_{yr} = 365 \cdot f_{fr} \cdot f_q$	$[g/year]$	3.65	3.65	-	-
Max consumption, c_{max}		$[g/\#N \cdot day]$	0.10	0.15	-	-
Efficiency, ε	$\varepsilon = \varphi/f_{yr}$	$[\#N/g]$	1.09	1.09	-	-
Clearance rate, b		$[m^2/\#N \cdot day]$	0.1e-5, 0.112e-5	0.1e-5, 0.115e-5	-	-
Guano, $\boldsymbol{R}$		$[\boldsymbol{g}/\boldsymbol{m}^2]$				
Influx-outflux rate, r		$[1/day]$	-	-	10/365	8/365
Carrying capacity, K		$[g/m^2]$	-	-	10000	10000

Case 2: Results

Initial parameters:
$R_{10} = 0$
$R_{20} = 0$
$N_{10} = 50$
$N_{10} = 50$

Both beetles have a higher clearance rate b for resource 2, and we do see that this resource 'stabilizes' at lower values. Again, N2 is expected to win the competition eventually.



Case 2: Code in Python

```
def deriv2(state, t, *params):
    eps, bs, ds, c_max, rs, K, nR = params
    Rs, Ns = state[:nR], state[nR:]

    R_char = lambda j: np.sum([bs[j][i]*Rs[i] for i in range(len(Rs))])

    C = lambda j: R_char(j)*c_max[j]/(R_char(j) + c_max[j])

    dR_dt = np.zeros(len(Rs))
    for iR, R in enumerate(Rs):
        dR_dt[iR] = rs[iR]*(K - R) - sum(C(j)*Ns[j] for j in range(len(Ns)))

    dN_dt = np.zeros(len(Ns))
    for iN, N in enumerate(Ns):
        dN_dt[iN] = (eps[iN]*C(iN) - ds[iN])*N

    return np.array([*dR_dt, *dN_dt])



bs = np.array([[0.1, 0.112],[0.1, 0.115]]) # sqm /(day*beetle)
bs = np.multiply(bs, 1e-5)
c_maxs = [0.1, 0.15] # g/(beetle*day)
ds = [1/yr, 1/yr]
rs = [10/yr, 8/yr]  # Unit: [1/day]
K = 10000 # Unit: [g/sqm]
t = np.linspace(0, 15000, 100000)
init_state = [0, 0, 50, 50]
nR = 2

params = (eps, bs, ds, c_maxs, rs, K, nR )
sol = odeint(deriv2, init_state, t, params).T
R1, R2, N1, N2 = sol
```

```
fig, ax = plt.subplots(1,1, figsize=(12, 6), tight_layout=True)
ax_twin = ax.twinx()
ax_twin.plot(t, R1 , label="R1", alpha=0.8, color='mediumseagreen', linestyle="--")
ax_twin.plot(t, R2 , label="R2", alpha=0.8, color = 'skyblue', linestyle="--")
ax.plot([], [] , label="R1", alpha=0.8, color='mediumseagreen', linestyle="--")
ax.plot([], [] , label="R2", alpha=0.8, color = 'skyblue', linestyle="--")

ax.plot(t, N1, label="N1", color='mediumvioletred', alpha=0.8)#, color =
'mediumseagreen', linestyle="--")
ax.plot(t, N2 , label="N2", color='rebeccapurple', alpha=0.8)#, color =
'mediumseagreen', linestyle="--")

ax.set_ylabel('Population density $\\left[ m^{-2} \\right]$')
ax_twin.set_ylabel('Guano density $\\left[ \\frac{g}{m^{2}} \\right]$')
ax.set_xlabel('Time [days]')

ax.set_ylim(bottom=0)
ax_twin.set_ylim(bottom=0)
ax.legend(bbox_to_anchor=(1.15, 1), loc=2, borderaxespad=0.5, fontsize=15)
```