

Hubbel assignment 1

Clive Lau

February 29, 2016

```
library(deSolve)
```

```
##  
## Attaching package: 'deSolve'  
##  
## The following object is masked from 'package:graphics':  
##  
##      matplot
```

Experiment 1: equal r and D, K_S varies

sp.1: C-8; sp.2: PAO

```
###Parameters (time scale in seconds)  
S_0 <- 1e-4  
D <- (6.0e-2)/3600  
  
y1 <- 2.5e10  
K_S1 <- 3.0e-6  
mu1 <- 0.81/3600  
r1 <- 0.75/3600  
J1 <- 2.4e-7  
  
y2 <- 3.8e10  
K_S2 <- 3.1e-4  
mu2 <- 0.91/3600  
r2 <- 0.85/3600  
J2 <- 2.19e-5  
  
parameters <- c(S_0=S_0, D=D, y1=y1, K_S1=K_S1, mu1=mu1, r1=r1, J1=J1, y2=y2, K_S2=K_S2, mu2=mu2, r2=r2)  
  
NutriODE <- function(tt, init, parameters) {  
  SS <- init[1]  
  N1 <- init[2]  
  N2 <- init[3]  
  with(as.list(parameters), {  
    dS.dt <- (S_0-SS)*D - mu1*SS*N1/(y1*(K_S1+SS)) - mu2*SS*N2/(y2*(K_S2+SS))  
    dN1.dt <- mu1*SS*N1/(K_S1+SS) - D*N1  
    dN2.dt <- mu2*SS*N2/(K_S2+SS) - D*N2  
    return(list(c(dS.dt, dN1.dt, dN2.dt)))  
  })  
}  
  
init <- c(SS=S_0, N1=1e3, N2=200*1e3)
```

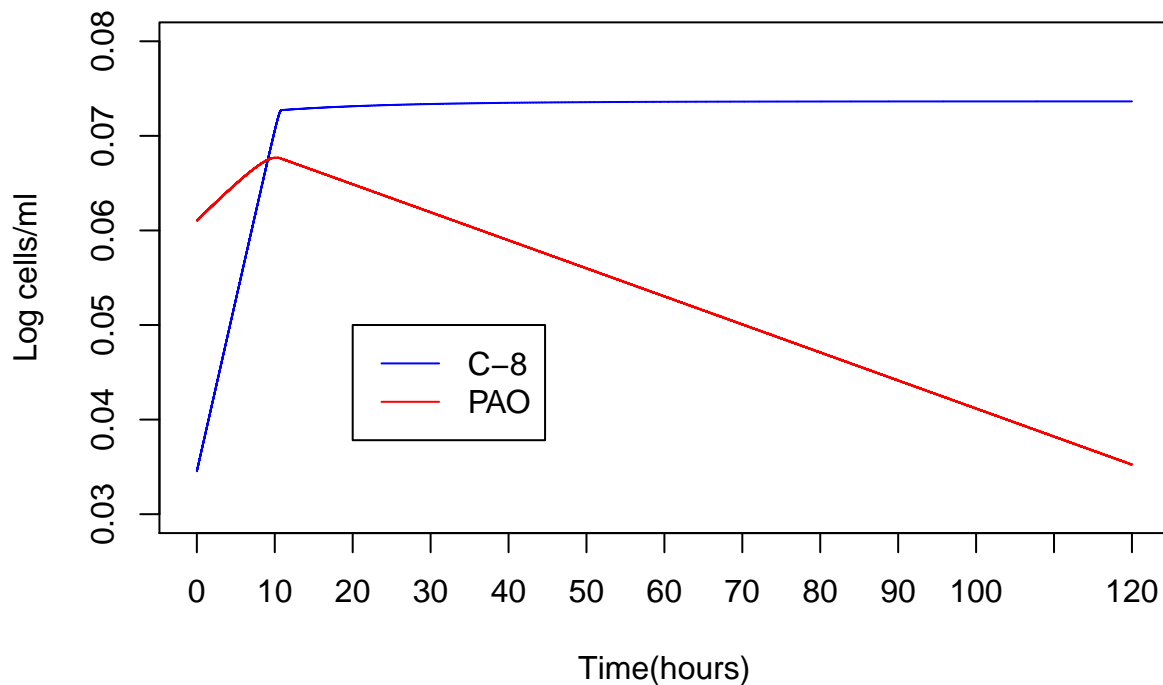
```

tseq <- seq(0, 120*3600)

exper1.out <- lsoda( init, tseq, NutriODE, parameters)

plot(exper1.out[,1]/3600, log(exper1.out[,3])/200, col="blue", type="l", ylim = c(0.03, 0.08), xaxt = "n")
lines(exper1.out[,1]/3600, log(exper1.out[,4])/200, col="red", type="l")
axis(side = 1, at = seq(0, length(exper1.out[,1])/3600, by=10))
legend(20, 0.05, c("C-8", "PA0"), col = c("blue", "red"), lty = c(1, 1))

```



Experiment 2: equal K_S and D , r varies

sp.1: C-8 nal(r)spec(s); sp.2: C-8 nal(s)spec(r)

```

S_0 <- 5e-6
D <- (7.5e-2)/3600

y1 <- 6.3e10
K_S1 <- 1.6e-6
mu1 <- 0.68/3600
r1 <- 0.61/3600
J1 <- 1.98e-7

```

```

y2 <- 6.2e10
K_S2 <- 1.6e-6
mu2 <- 0.96/3600
r2 <- 0.89/3600
J2 <- 1.35e-7

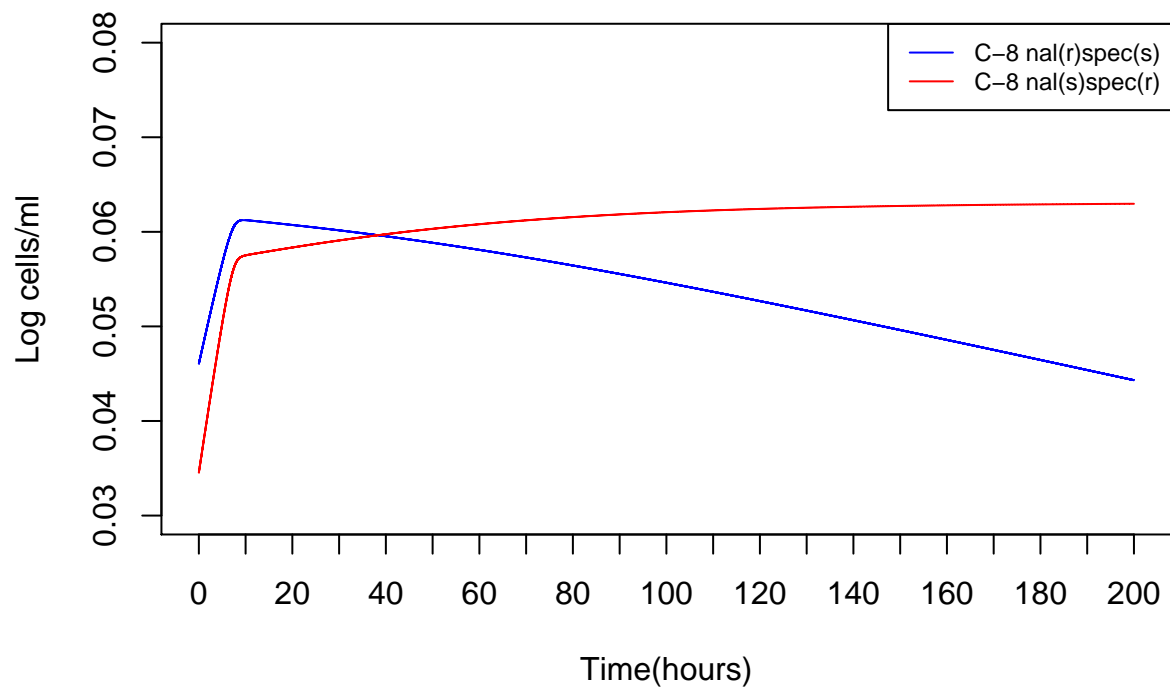
init <- c(SS=S_0, N1=100*1e2, N2=1e3)
tseq <- seq(0, 200*3600)

parameters <- c(S_0=S_0, D=D, y1=y1, K_S1=K_S1, mu1=mu1, r1=r1, J1=J1, y2=y2, K_S2=K_S2, mu2=mu2, r2=r2)

exper2.out <- lsoda( init, tseq, NutriODE, parameters)

plot(exper2.out[,1]/3600, log(exper2.out[,3])/200, col="blue", type="l", ylim = c(0.03, 0.08), xaxt = "n",
lines(exper2.out[,1]/3600, log(exper2.out[,4])/200, col="red", type="l")
axis(side = 1, at = seq(0, length(exper2.out[,1])/3600, by=10))
legend("topright", c("C-8 nal(r)spec(s)", "C-8 nal(s)spec(r)"), col = c("blue", "red"), lty = c(1, 1), cex=1.2)

```



Experiment 3: equal J, K_S and r vary

sp.1: C-8 nal(r)spec(s); sp.2: C-8 nal(s)spec(r)

```
S_0 <- 5e-6
D <- (7.5e-2)/3600

y1 <- 6.3e10
K_S1 <- 1.6e-6
mu1 <- 0.68/3600
r1 <- 0.61/3600
J1 <- 1.98e-7

y2 <- 6.2e10
K_S2 <- 0.9e-6
mu2 <- 0.41/3600
r2 <- 0.34/3600
J2 <- 1.99e-7

init <- c(SS=S_0, N1=100*1e2, N2=1e3)
tseq <- seq(0, 120*3600)

parameters <- c(S_0=S_0, D=D, y1=y1, K_S1=K_S1, mu1=mu1, r1=r1, J1=J1, y2=y2, K_S2=K_S2, mu2=mu2, r2=r2)

exper3.out <- lsoda( init, tseq, NutriODE, parameters)

plot(exper3.out[,1]/3600, log(exper3.out[,3])/200, col="blue", type="l", ylim = c(0.03, 0.08), xaxt = "n")
lines(exper3.out[,1]/3600, log(exper3.out[,4])/200, col="red", type="l")
axis(side = 1, at = seq(0, length(exper3.out[,1])/3600, by=10))
legend("topright", c("C-8 nal(r)spec(s)", "C-8 nal(s)spec(r)"), col = c("blue", "red"), lty = c(1, 1), cex = 1.2)
```

