SIP - simple foodweb

LP

29-11-2021

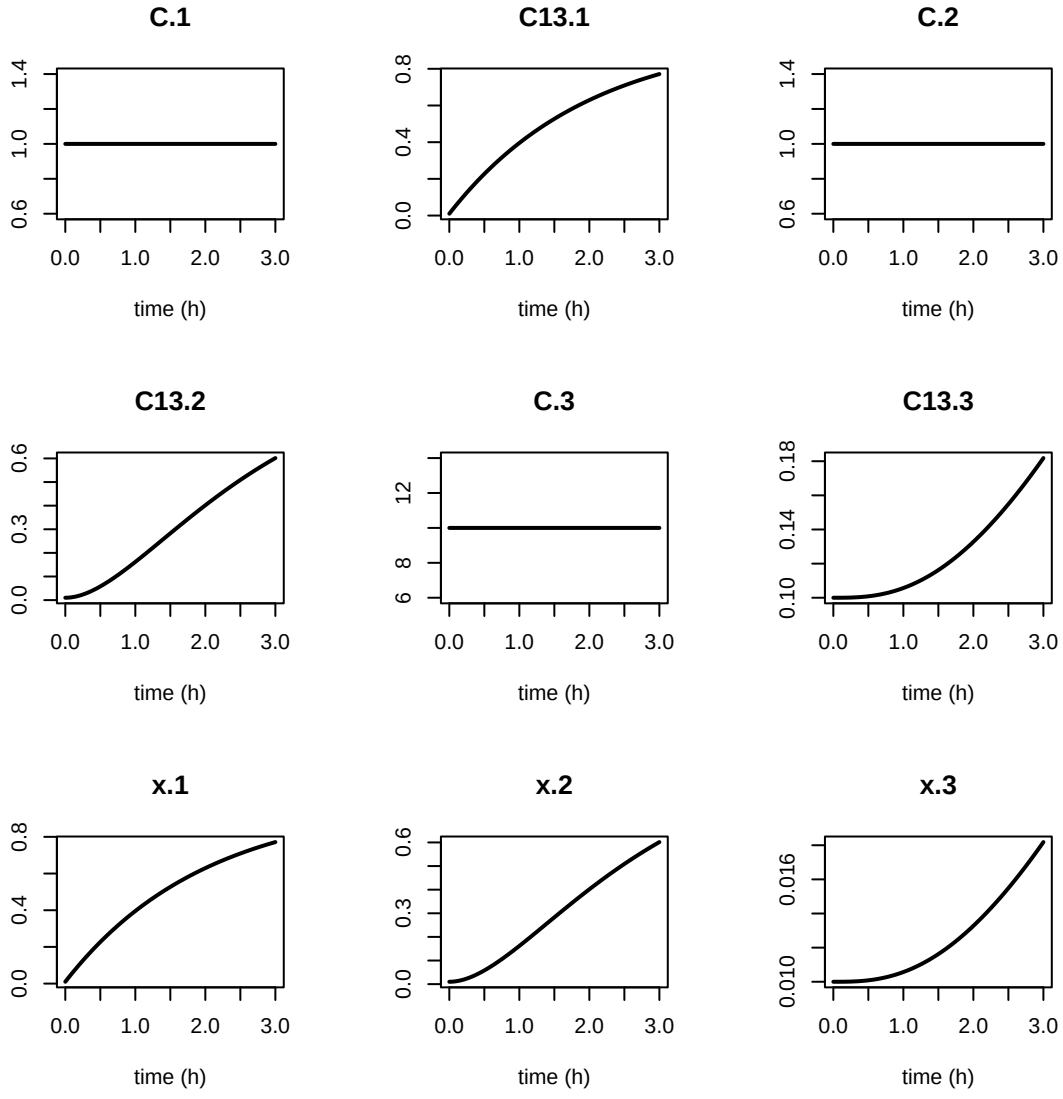
Contents

```
scenario <- 1

C1.ini <- 1
C2.ini <- 1
C3.ini <- 10
r <- 1
x.ini <- 0.01
tmax <- C1.ini/r*3

# Model parameters
pars <- c(
  r.1   = r/2,
  r.2   = r/1,
  r.3   = r/10,
  x.S   = 0.99
)

# Model function: calculates time-derivatives and other output
C13model <-function(t, state, pars) {
  # t: time, state: state variables, pars: model parameters
  with (as.list(c(state, pars)),{

  x.1  <- C13.1/C.1
  x.2  <- C13.2/C.2
  x.3  <- C13.3/C.3

  # simultaneous synthesis and breakdown
  if (scenario==1){
    dCdt.1   <- r.1     - r.1
    dC13dt.1 <- r.1*x.S - r.1*x.1
    dCdt.2   <- r.2     - r.2
    dC13dt.2 <- r.2*x.1 - r.2*x.2
    dCdt.3   <- r.3     - r.3
    dC13dt.3 <- r.3*x.2 - r.3*x.3
  }

  list(
    c(dCdt.1, dC13dt.1, dCdt.2, dC13dt.2, dCdt.3, dC13dt.3),
    x.1 = x.1,
    x.2 = x.2,
```

```
    x.3 = x.3
      )
  })
}
```

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

# integrate the model with the new parameters
outtimes <- seq(from = 0, to = tmax, length.out = 100)
yini <- c(C.1 = C1.ini, C13.1 = C1.ini*x.ini,
          C.2 = C2.ini, C13.2 = C2.ini*x.ini,
          C.3 = C3.ini, C13.3 = C3.ini*x.ini)
out <- ode(y = yini, parms = pars, func = C13model, times = outtimes)
```

```
plot(out, xlab="time (h)", lty=1, lwd=2, mfrow=c(3,3))
```




```
par(mfrow=c(2,2))
```

```
matplot(outtimes, cbind(out[,"x.1"], out[,"x.2"], out[,"x.3"]),
        xlab="time (h)", ylab="x1, x2, x3", type="l", lty=1, lwd=2)
plot(out[,"x.1"], out[,"x.2"], type="l", lty=1, lwd=2)
lines(c(0,1), c(0,1), lty=2)
plot(out[,"x.2"], out[,"x.3"], type="l", lty=1, lwd=2)
lines(c(0,1), c(0,1), lty=2)
plot(out[,"x.1"], out[,"x.3"], type="l", lty=1, lwd=2)
lines(c(0,1), c(0,1), lty=2)
```