Final project

Juan Carlos Villaseñor-Derbez

2018-03-15

$$
\begin{aligned}
\frac{dX}{dt} &= r_x X \left(1 - \frac{X}{K_x}\right) - a_x P X - h_x X \\
\frac{dP}{dt} &= [c(a_x X + a_y Y) - d_P] P \left(1 - \frac{P}{K_p}\right) - h_p P
\end{aligned}
\tag{1}
$$

```r
pred_prey <- function(t, values, pars){

  # Extract parameters
  rx <- pars[1]
  Kx <- pars[2]
  ax <- pars[3]
  hx <- pars[4]
  c <- pars[5]
  ay <- pars[6]
  dp <- pars[7]
  Kp <- pars[8]
  hp <- pars[9]

  Y <- 500

  # Extract state variables
  X <- values[1]
  P <- values[2]

  # Update equations

  dXdt <- (rx*X)*(1-(X/Kx))-(ax*P*X)-(hx*X)
  dPdt <- P*((c*((ax*X)+(ay*Y)))-dp)*(1-(P/Kp))-(hp*P)

  return(list(c(dXdt, dPdt)))
}
```

```r
run_model <- function(rx = 1, Kx = 100, ax = 0.03, hx = 0.65, c = 0.05, ay = 0.03, dp = 0.25, Kp = 25, 
  # Define parameters
  pars <- c(rx, Kx, ax, hx, c, ay, dp, Kp, hp)

  # Define values
  values <- c(X0, P0)

  # Define time
  time <- seq(t0, t1, by = 0.1)

  lsoda(y = values, times = time, func = pred_prey, parms = pars) %>%
    as.data.frame() %>%
    magrittr::set_colnames(value = c("Time", "X", "P")) %>%
    gather(Organism, Abundance, X, P) %>%
    mutate(H = case_when(Organism == "X" ~ hx*Abundance,
                         Organism == "P" ~ hp*Abundance))
}
```

```r
extract_state_vars <- function(scenario){
  last_values <- scenario %>%
    filter(Time == max(Time)) %>%
    group_by(Organism) %>%
    select(-H) %>%
    spread(Organism, Abundance)

  return(c(last_values$X, last_values$P))
  }
```

Synchronous recovery

```r
start <- run_model(hx = 0, hp = 0, t1 = 30, X0 = 25, P0 = 25)

c(X1, P1) %<-% extract_state_vars(start)
harvest_P <- run_model(hx = 0, t0 = 30, t1 = 70, X0 = X1, P0 = P1)

c(X2, P2) %<-% extract_state_vars(harvest_P)
harvest_X <- run_model(t0 = 70, t1 = 120, X0 = X2, P0 = P2)

c(X3, P3) %<-% extract_state_vars(harvest_X)
stop_harvest <- run_model(hx = 0, hp = 0, t0 = 120, t1 = 200, X0 = X3, P0 = P3)


both <- rbind(start, harvest_P, harvest_X, stop_harvest) %>%
  ggplot(aes(x = Time, y = Abundance)) +
  geom_line(aes(color = Organism), size = 1) +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "none")

both_h <- rbind(start, harvest_P, harvest_X, stop_harvest) %>%
  ggplot(aes(x = Time, y = H)) +
  geom_line(aes(color = Organism), size = 1) +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "none")
```

Predator first

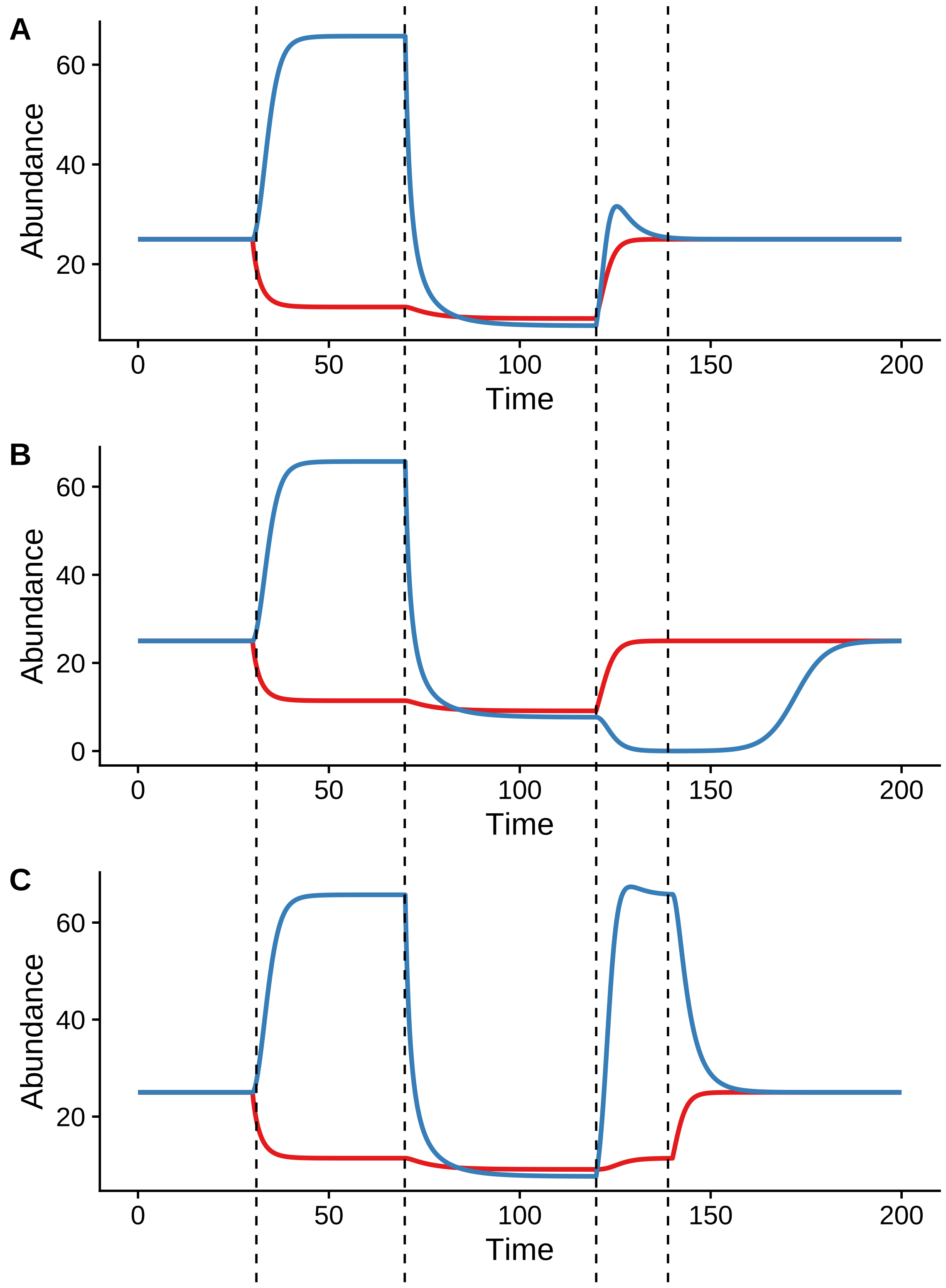
```r
c(X3, P3) %<-% extract_state_vars(harvest_X)
stop_harvest_P <- run_model(hp = 0, t0 = 120, t1 = 140, X0 = X3, P0 = P3)

c(X4, P4) %<-% extract_state_vars(stop_harvest_P)
stop_harvest_X <- run_model(hx = 0, hp = 0, t0 = 140, t1 = 200, X0 = X4, P0 = P4)

p_first <- rbind(start, harvest_P, harvest_X, stop_harvest_P, stop_harvest_X) %>%
  ggplot(aes(x = Time, y = Abundance)) +
  geom_line(aes(color = Organism), size = 1) +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "none")

p_first_h <- rbind(start, harvest_P, harvest_X, stop_harvest_P, stop_harvest_X) %>%
  ggplot(aes(x = Time, y = H)) +
  geom_line(aes(color = Organism), size = 1) +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "none")
```

Prey first

```
c(X3, P3) %<-% extract_state_vars(harvest_X)
stop_harvest_X <- run_model(hx = 0, t0 = 120, t1 = 140, X0 = X3, P0 = P3)

c(X4, P4) %<-% extract_state_vars(stop_harvest_X)
stop_harvest_P <- run_model(hp = 0, hx = 0, t0 = 140, t1 = 200, X0 = X4, P0 = P4)

x_first <- rbind(start, harvest_P, harvest_X, stop_harvest_X, stop_harvest_P) %>%
  ggplot(aes(x = Time, y = Abundance)) +
  geom_line(aes(color = Organism), size = 1) +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "none")

x_first_h <- rbind(start, harvest_P, harvest_X, stop_harvest_X, stop_harvest_P) %>%
  ggplot(aes(x = Time, y = H)) +
  geom_line(aes(color = Organism), size = 1) +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "none")
```

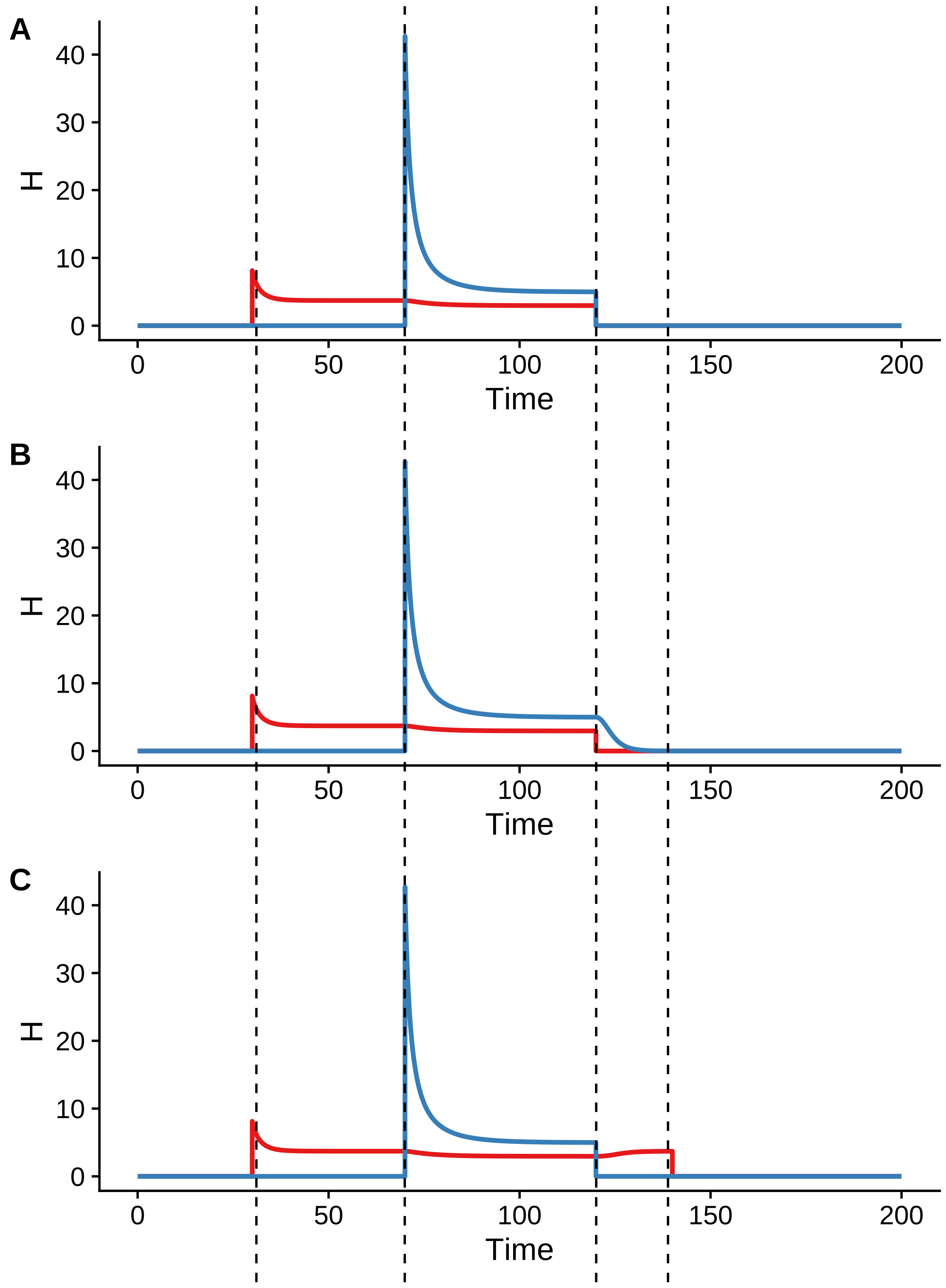
Visualize scenarios

```
plot_grid(both, p_first, x_first, labels = "AUTO", ncol = 1) +
  geom_vline(xintercept = c(0.27, 0.425, 0.625, 0.7), linetype = "dashed")
```



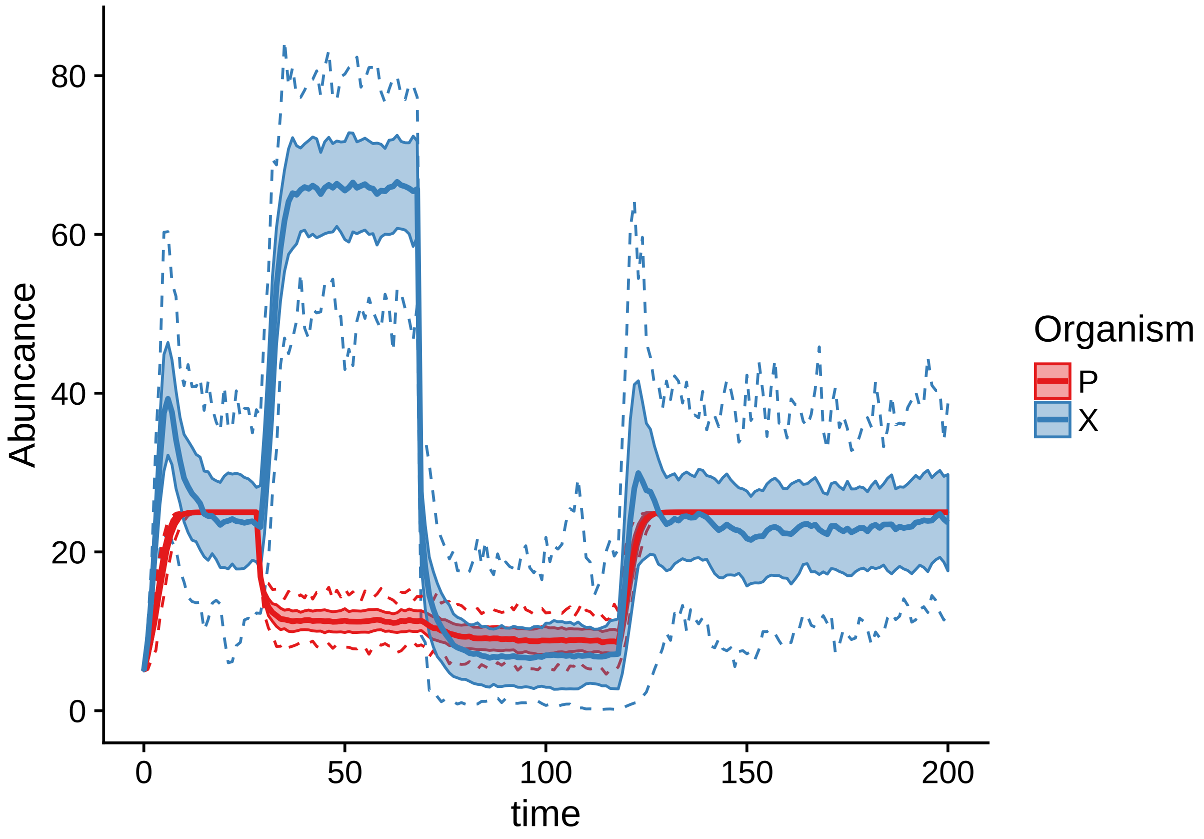
```
# ggsave(filename = "all.png", width = 6, height = 4.2)
```

```
plot_grid(both_h, p_first_h, x_first_h, labels = "AUTO", ncol = 1) +
  geom_vline(xintercept = c(0.27, 0.425, 0.625, 0.7), linetype = "dashed")
```



```
# Define parameters
rx <- 1
Kx <- 100
ax <- 0.03
hx <- 0.65
```

```r
c <- 0.05
ay <- 0.03
dp <- 0.25
Kp <- 25
hp <- 0.325

X0 <- 5
P0 <- 5
set.seed(43)
nsteps = 200

all_results <- tibble(time = NA, X = NA, P = NA)

for(j in 1:100){
  time <- seq(0, nsteps, by = 1)
  X <- rep(0, length(time))
  P <- X

  X[1] <- X0
  P[1] <- P0

  for(i in 2:(nsteps+1)){

    hx <- case_when(i < 30 ~ 0,
                    i >= 30 & i < 70 ~ 0,
                    i >= 70 & i < 120 ~ 0.65,
                    i >= 120 ~ 0)

    hp <- case_when(i < 30 ~ 0,
                    i >= 30 & i < 70 ~ 0.325,
                    i >= 70 & i < 120 ~ 0.325,
                    i >= 120 ~ 0)

    rx_s <- rnorm(n = 1, mean = rx, sd = 0.2)
    Y <- rnorm(n = 1, mean = 500, sd = 100)

    X[i] <- X[i-1] + rx_s*X[i-1]*(1-(X[i-1]/Kx))-(ax*P[i-1]*X[i-1])-(hx*X[i-1])
    P[i] <- P[i-1] + (c*((ax*X[i-1])+(ay*Y))-dp)*P[i-1]*(1-(P[i-1]/Kp)) - (hp*P[i-1])

  }

  results_i <- tibble(time, X = X, P = P)
  all_results <- rbind(all_results, results_i)
}


all_results %>%
  filter(!is.na(time)) %>%
  gather(Organism, Abundance, X, P) %>%
  group_by(time, Organism) %>%
  summarize(minA = min(Abundance),
            maxA = max(Abundance),
            sdA = sd(Abundance),
            meanA = mean(Abundance)) %>%
  ungroup() %>%
  ggplot(aes(x = time, color = Organism, fill = Organism)) +
  geom_line(aes(y = minA), linetype = "dashed") +
  geom_line(aes(y = maxA), linetype = "dashed") +
  geom_ribbon(aes(ymin = meanA - sdA, ymax = meanA + sdA), alpha = 0.4) +
  geom_line(aes(y = meanA), size = 1) +
  scale_color_brewer(palette = "Set1") +
  scale_fill_brewer(palette = "Set1") +
  labs(y = "Abuncance")
```


Abuncance
time
Organism
P
X