QBIO7004

jasmine fowler-morrow

2022-05-25

Introduction

The abundance of aquatic vegetation (macrophytes) in shallow lakes play a key role in reducing turbidity and providing a food source for inhabitants. However, disturbing or removing macrophytes can cause rapid and irreversible changes to conditions (schaffer, 1998). Such observations have led researchers to conclude that shallow lakes usually have two stable states; clear and turbid (scheffer, 1993). Furthermore, the state of the habitat primarily depends on the macrophyte population (ref).

Water mixing and light levels mean shallow lakes are rich in species. However, like many other aquatic ecosystems, the size and quality of such habitats has declined (schaffer).

The occurence of tipping points in this ecosystem has been studied extensively (references). trait variation and evolution are important for understanding tipping point dynamics. Dakos et al (2019) found that phenotypic variation can make macrophytes more resilient to collapse.

Methods

I created an individual-based stochastic model tracking macrophyte population dynamics over time in response to turbidity.

```
library(tidyverse)
library(reshape2)
#install.packages("patchwork")
library(patchwork)
```

```
##Simulation function
#Inputs:
#inipop = dataframe of initial macrophyte population
#initurb = number describing the starting turbidity levels
#p = list of parameters
#returns: macrophyte number, turbidity level and trait (z) mean and sd
macrophyte_growth <- function(inipop, initurb, tmax, p){
  ###set up data###
  Turb <- rep(0, tmax) #store turbidity over simulation
  Turb[1] <- initurb #initial Turbidity level

  #macrophyte population
  M <- rep(0, tmax) #store population size over simulation
  M[1] <- length(inipop$ID) #initial population size
  mtt <- as.vector(rep(NA, tmax), "list") #macrophytes through time
  mtt[[1]] <- inipop
```

```
#trait value summaries
Z_means <- rep(0, tmax)
Z_means[1] <- mean(inipop$z)
Z_sd <- rep(0, tmax)
Z_sd[1] <- sd(inipop$z)

for (i in 1:(tmax-1)){
  #create empty data frame with empty rows for reproduction
  mtt[[i+1]] <- data.frame(ID = rep(NA,p$K +20),
                           z = NA,
                           alive = NA)
  mtt[[i+1]][1:nrow(mtt[[i]]),] <- mtt[[i]] #copy population over to next time step

  #update turbidity level
  Turb[i+1] <- Turb[i] +
    p$rt*Turb[i]*(1-(Turb[i]/(p$T0*(p$hm/(p$hm +(M[i]/p$K))))))

  #determine the outcome of each individual in this time-step
  for (j in 1:length(mtt[[i+1]]$ID[!is.na(mtt[[i + 1]]$ID)])){
    #each individual has probability of death
    mtt[[i+1]]$alive[j] <- rbernoulli(1,1-p$mu)

    if (mtt[[i+1]]$alive[j]){
      #if alive there is a chance of asexual reproduction
      #ht value for that individual
      ht <- exp(0.5*mtt[[i+1]]$z[j])
      reproduce <- rbernoulli(1,
                              p$rm*M[i]*(1-(M[i]/p$K)*((ht^4+Turb[i+1])/ht^4)))

      if (reproduce) { #create new individual
        emptySlot <- which(is.na(mtt[[i + 1]]$ID))[1] # next empty row in dataframe
        mtt[[i+1]]$ID[emptySlot] <- last(mtt[[i+1]]$ID[!is.na(mtt[[i + 1]]$ID)]) + 1
        mtt[[i+1]]$z[emptySlot] <- mtt[[i+1]]$z[j] + rnorm(1,sd = p$sigma)
        mtt[[i+1]]$alive[emptySlot] <- TRUE

      }
    }
  }
  #remove rows with na's
  mtt[[i + 1]] <- mtt[[i + 1]][!is.na(mtt[[i + 1]]$ID),]
  #remove dead individuals
  mtt[[i + 1]] <- mtt[[i + 1]][mtt[[i + 1]]$alive,]
  #new macrophyte population size
  M[i+1] <- length(mtt[[i+1]]$ID)

  # check whether population is extinct
  if (M[i+1]==0){
    return(list(M = M, Turb = Turb, Z_means = Z_means, Z_sd = Z_sd))
    stop("Macrophyte population collapse")
  }

  #otherwise, update population information
  else{
```

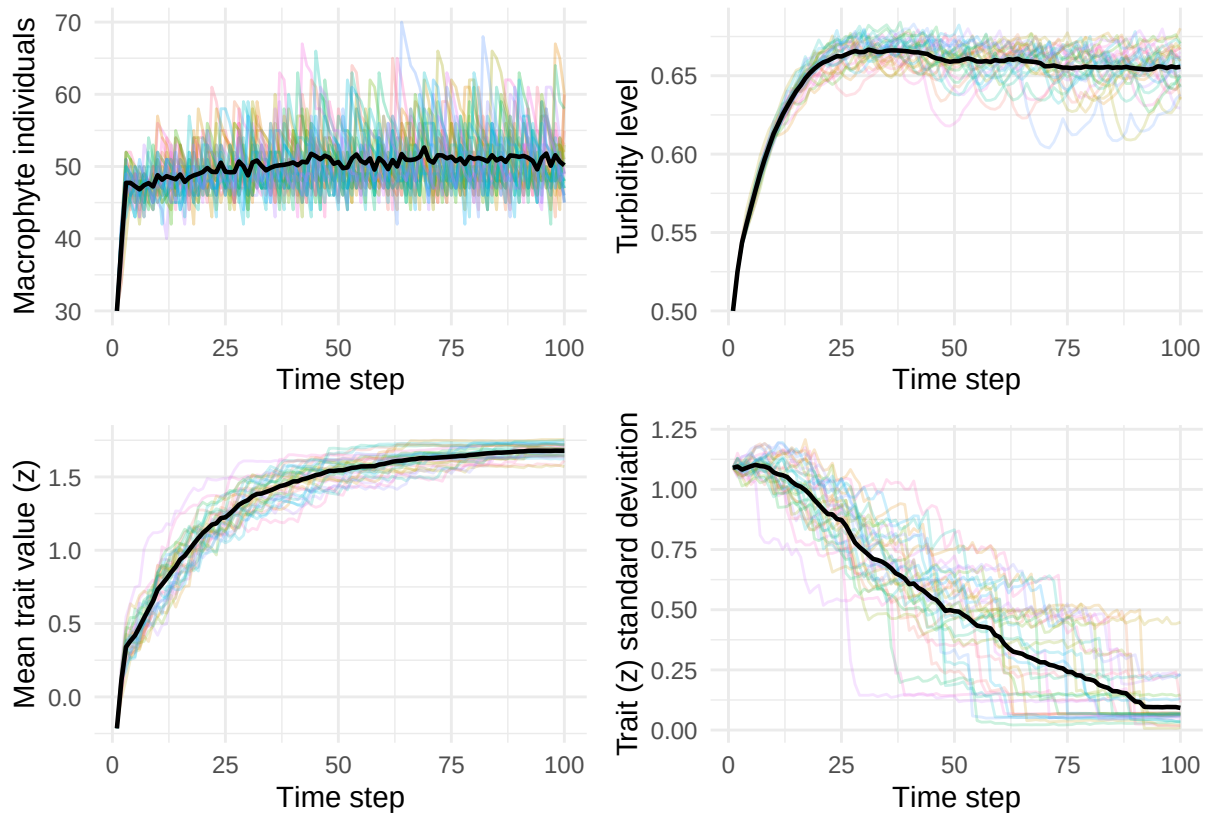
```
      Z_means[i+1] <- mean(mtt[[i+1]]$z) #new population trait val mean
      Z_sd[i+1] <- sd(mtt[[i+1]]$z) #new population trait sd
    }

  }
  return(list(M = M, Turb = Turb, Z_means = Z_means, Z_sd = Z_sd))
}
```

```
######## Running function ############
#this function runs the macrophyte population model numerous times
#it allows parameters and initial values to be specified
# it requires number of simulations
#returns 4 matrices corresponding to
runMacrophyte <- function(n_sims, tmax = 100, rm = 0.1, rt = 0.1,
                          hm = 0.2, T0 = 3, zmin = -2, zmax = 2,
                          K = 50, mu = 0.05, sigma = 0.001, n0 = 5,initurb = 0.5){
  #create parameter list
  p <- list() #parameter list
  p$rm <- rm #macrophyte growth rate
  p$rt <- rt #turbidity growth rate
  p$hm <- hm #macrophyte half saturation
  p$T0 <- T0 #background turbidity
  p$K <- K #macrophyte carrying capacity
  p$mu <- mu #probability of death
  p$sigma <- sigma

  #create data frame for initial population
  inipop <- data.frame(ID = 1:n0,
                       z = runif(n0, zmin, zmax), #generate trait values
                       alive = TRUE)

  ##matrices to store simulation results
  mac_pop <- matrix(nrow = n_sims,ncol = tmax)
  turb <- matrix(nrow = n_sims,ncol = tmax)
  Z_means <- matrix(nrow = n_sims,ncol = tmax)
  Z_sd <- matrix(nrow = n_sims,ncol = tmax)
  ##run simulations
  for (i in 1:n_sims){
    results <- macrophyte_growth(inipop, initurb, tmax, p)
    mac_pop[i,] <- results$M
    turb[i,] <- results$Turb
    Z_means[i,] <- results$Z_means
    Z_sd[i,] <- results$Z_sd
  }
  return(list(mac_pop = mac_pop,turb = turb, Z_means = Z_means, Z_sd = Z_sd))
}
```

```
#this function summarises the simulation results
#summary = "plot" returns a plot of the data over time
Macrophyte_sim_summary <- function(result, summary = "plot"){
  ### MACROPHYTES ###
  df <- melt(result$mac_pop, varnames = c("Sim","Time"), value.name = "M")
  dfmean <- df %>%
```

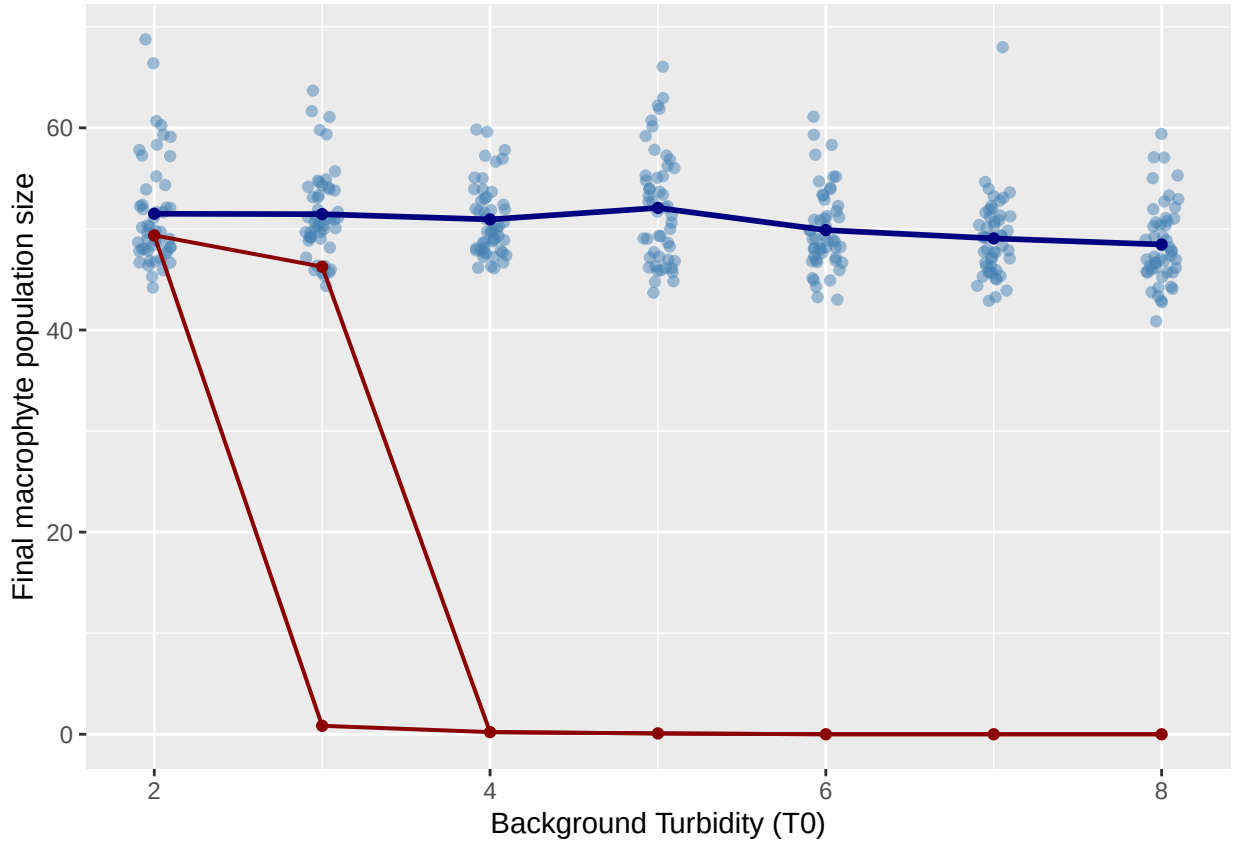
```
  group_by(Time) %>%
  summarise(mean_M = mean(M), se = sd(M)/length(M))

### TURBIDITY ###
df2 <- melt(result$turb, varnames = c("Sim","Time"), value.name = "Tb")
df2mean <- df2 %>%
  group_by(Time) %>%
  summarise(mean_T = mean(Tb))

### TRAIT MEAN ###
df3 <- melt(result$Z_means, varnames = c("Sim","Time"), value.name = "zbar")
df3mean <- df3 %>%
  group_by(Time) %>%
  summarise(mean_z = mean(zbar))

### TRAIT SD ###
df4 <- melt(result$Z_sd, varnames = c("Sim","Time"), value.name = "zsd")
df4mean <- df4 %>%
  group_by(Time) %>%
  summarise(mean_zsd = mean(zsd))

if (summary == "plot"){
    p1 <- ggplot() +
      geom_line(data = df, aes(x = Time, y = M, group = Sim,
                               colour = as.character(Sim)),
            show.legend = FALSE, alpha = 0.3)+
      geom_line(data = dfmean, aes(x = Time, y = mean_M), size = 0.8) +
      theme_minimal() + labs(x = "Time step", y = "Macrophyte individuals")
    p2 <- ggplot() +
        geom_line(data = df2, aes(x = Time, y = Tb,
                                  group = Sim, colour = as.character(Sim)),
                  show.legend = FALSE, alpha = 0.2)+
        geom_line(data = df2mean, aes(x = Time, y = mean_T), size = 0.8) +
        theme_minimal()+ labs(x = "Time step", y = "Turbidity level")
    p3 <- ggplot() +
      geom_line(data = df3, aes(x = Time, y = zbar, group = Sim,
                               colour = as.character(Sim)),
            show.legend = FALSE, alpha = 0.2)+
      geom_line(data = df3mean, aes(x = Time, y = mean_z), size = 0.8) +
      theme_minimal()+ labs(x = "Time step", y = "Mean trait value (z)")
    p4 <- ggplot() +
      geom_line(data = df4, aes(x = Time, y = zsd, group = Sim,
                               colour = as.character(Sim)),
            show.legend = FALSE, alpha = 0.2)+
      geom_line(data = df4mean, aes(x = Time, y = mean_zsd), size = 0.8) +
      theme_minimal()+ labs(x = "Time step", y = "Trait (z) standard deviation")
    ### use patchwork to return ###
    p <- (p1 | p2)/ (p3 | p4)
  return(p)
}
else if (summary == "stats"){
  M_end <- dfmean[dfmean$Time==100,2]
  Turb_end <- df2mean[df2mean$Time==100,2]
```

```
    return(data.frame(M_end = M_end, Turb_end = Turb_end))
  }
  else if(summary == "data"){
    return(list(M = df, Tb = df2, z = df3))
  }

}

result <- runMacrophyte(30, initurb = 0.5, T0 = 4, n0 = 30)
p <- Macrophyte_sim_summary(result)
p
```




```
M <- rep(0,7)
final_mac <- data.frame(T0 = NULL, final_M = NULL)
for (t in 2:8){
  #run simulation for given T0 value
  result <- runMacrophyte(50, initurb = 0.5, T0 = t, n0 = 10)

  #average final macrophyte population size
  M[t-1] <- Macrophyte_sim_summary(result, "stats")[[1]]

  #final macrophyte population size for each simulation
  Mac <- Macrophyte_sim_summary(result, summary = "data")[[1]]
  Mac <- cbind(T0 = rep(t,10), final_M = Mac[Mac$Time==100,"M"])
  #store final size with T0 level
```

```
  final_mac <- rbind(final_mac, Mac)

}
analytic_M <- read.csv("analytic_M.csv")


## plot ##
ggplot()+geom_jitter(data = final_mac, aes(x = T0, y = final_M),
                     col = "steelblue", alpha = 0.5, width = 0.1) +
  geom_line(aes(x = 2:8, y = M), col = "navy", lwd = 1) +
  geom_point(aes(x = 2:8, y = M), col = "navy") +
  labs(x = "Background Turbidity (T0)", y="Final macrophyte population size")+
  geom_line(data = analytic_M[-2,], aes(x = T0, y = M), col = "red4", lwd = 0.7) +
  geom_line(data = analytic_M[-3,], aes(x = T0, y = M), col = "red4", lwd = 0.7) +
  geom_point(data = analytic_M, aes(x = T0, y = M), col = "red4")
```



Conclusion