

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 commonly have two stable states; clear and turbid (scheffer, 1993). The state of the ecosystem will change after a “tipping point” is reached.

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 occurrence of tipping points in this ecosystem has been studied extensively (references). Previous research has found that whether or not this shift in states occurs depends on the macrophyte population size (ref). Dakos et al further argued that 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. However, this model was studied analytically with trait values remaining constant through time. For this reason, I wished to add stochasticity and individual variation of trait value.

Methods

I created an individual-based stochastic model tracking macrophyte population dynamics over time in response to turbidity. The model tracks a population of macrophyte individuals which each have a given trait value (z).

Each time period, an individual has a fixed probability of dying. Given an individual does not die, they have a probability of reproducing (eq 1). Higher turbidity decreases their chance of reproducing as does lower trait values, which influences their response to turbidity (eq 2).

It is assumed that all reproduction is asexual, and offspring receive the same trait value as their parent with some random genetic mutation.

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

Code

```
##### 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 (n_sims)
#returns 4 matrices corresponding to macrophyte population, turbidity level, mean trait value and trait
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 = 10, 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 = 2*(-0.5+rbeta(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)),

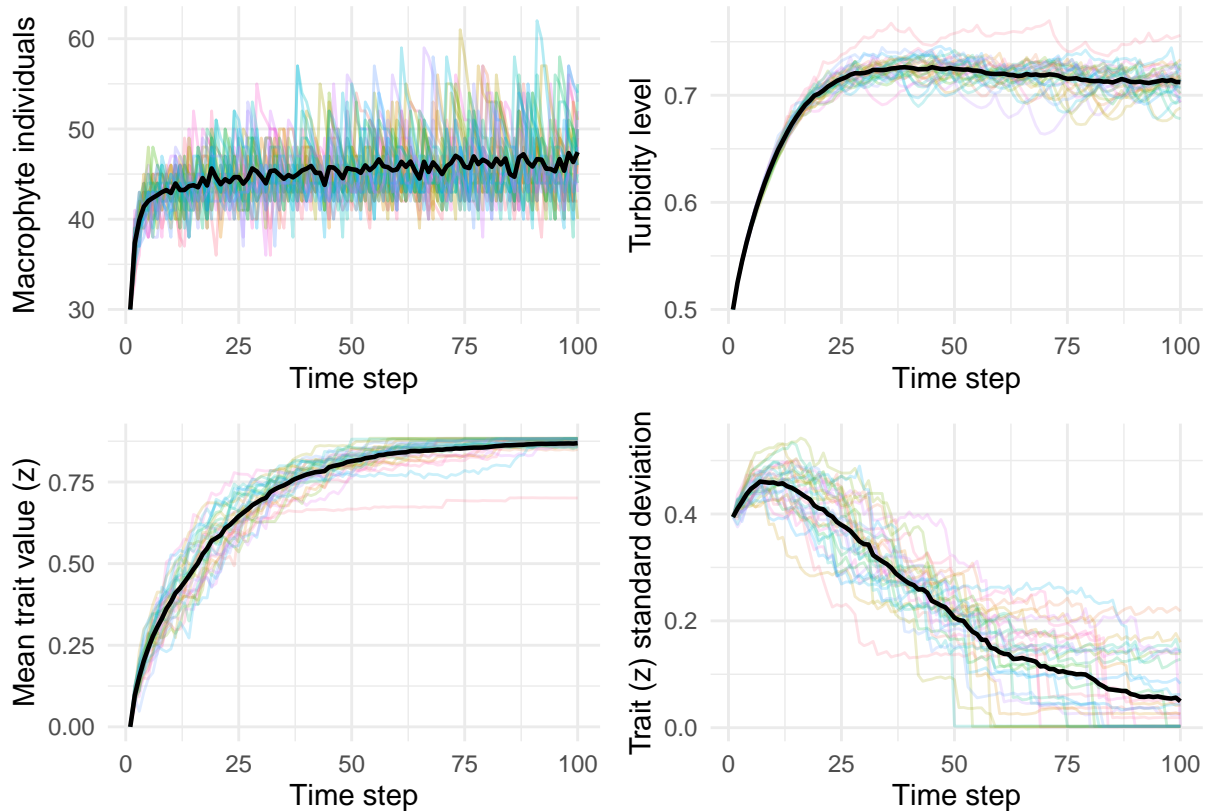
```

```

        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

```



```
#this function summarises the simulation results from different initial turbidity values
#summary = "plot" returns a plot of the data over time
vary_initial_turbidity <- function(){
```

```
  #initial turbidity
  turb <- c(0.1,0.3, 0.6, 1)
  macrophytes <- data.frame(NULL)
  turbidity <- data.frame(NULL)
  trait_val <- data.frame(NULL)
```

```
  for (t in 1:length(turb)){
    #simulate for turbidity value t
    result <- runMacrophyte(30, initurb = turb[t])
```

```
    ### MACROPHYTES ###
```

```
    df <- melt(result$mac_pop, varnames = c("Sim","Time"), value.name = "M") %>%
      group_by(Time) %>%
      summarise(mean_M = mean(M), se = sd(M)/sqrt(length(M))) %>%
      add_column(initurb = turb[t])
    macrophytes <- rbind(macrophytes, df)
```

```
    ### TURBIDITY ###
```

```
    df2 <- melt(result$turb, varnames = c("Sim","Time"), value.name = "Tb") %>%
      group_by(Time) %>%
      summarise(mean_T = mean(Tb), se = sd(Tb)/length(Tb)) %>%
      add_column(initurb = turb[t])
```

```

turbidity <- rbind(turbidity,df2)

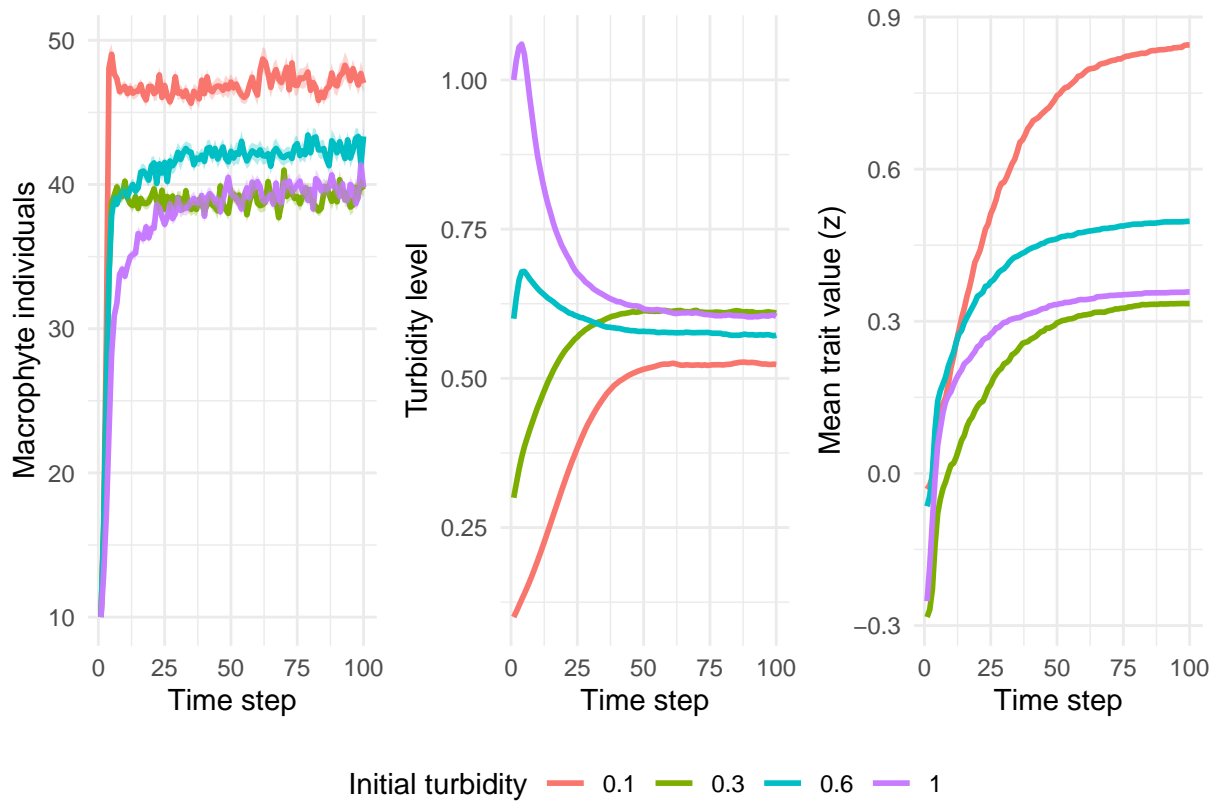
### TRAIT MEAN ###
df3 <- melt(result$Z_means, varnames = c("Sim","Time"), value.name = "zbar")%>%
  group_by(Time) %>%
  summarise(mean_z = mean(zbar), se = sd(zbar)/length(zbar))%>%
  add_column(initurb = turb[t])
trait_val <- rbind(trait_val,df3)
}

p1 <- ggplot() +
  geom_line(data = macrophytes, aes(x = Time, y = mean_M, group = initurb,
                                   colour = as.character(initurb)), lwd = 1) +
  geom_ribbon(data = macrophytes, aes(x = Time, ymin = mean_M-se, ymax = mean_M+se,
                                   group = initurb,fill = as.character(initurb)),
            alpha = 0.3, show.legend = FALSE)+
  theme_minimal() + labs(x = "Time step", y = "Macrophyte individuals", colour = "Initial turbidity")+
  theme(legend.position = "bottom")
p2 <- ggplot() +
  geom_line(data = turbidity, aes(x = Time, y = mean_T, group = initurb,
                                   colour = as.character(initurb)), lwd = 1) +
  geom_ribbon(data = turbidity, aes(x = Time, ymin = mean_T-se, ymax = mean_T+se,
                                   group = initurb,fill = as.character(initurb)),
            alpha = 0.3, show.legend = FALSE)+
  theme_minimal()+ labs(x = "Time step", y = "Turbidity level", colour = "Initial turbidity")+
  theme(legend.position = "bottom")
p3 <- ggplot() +
  geom_line(data = trait_val, aes(x = Time, y = mean_z, group = initurb,
                                   colour = as.character(initurb)), lwd = 1) +
  geom_ribbon(data = trait_val, aes(x = Time, ymin = mean_z-se, ymax = mean_z+se,
                                   group = initurb,fill = as.character(initurb)),
            alpha = 0.3, show.legend = FALSE)+
  theme_minimal()+ labs(x = "Time step", y = "Mean trait value (z)", colour = "Initial turbidity")+
  theme(legend.position = "bottom")

### use patchwork to return ###
p <- (p1 | p2 | p3) + plot_layout (guides = "collect") &
  theme(legend.position='bottom')
return(p)
}

vary_initial_turbidity()

```



```
vary_initial_mac <- function(){

  #initial turbidity
  size <- c(5,10, 20, 40)
  macrophytes <- data.frame(NULL)
  turbidity <- data.frame(NULL)
  trait_val <- data.frame(NULL)

  for (n in 1:length(size)){
    #simulate for turbidity value t
    result <- runMacrophyte(30, n0 = size[n])

    ### MACROPHYTES ###
    df <- melt(result$mac_pop, varnames = c("Sim","Time"), value.name = "M") %>%
      group_by(Time) %>%
      summarise(mean_M = mean(M), se = sd(M)/sqrt(length(M))) %>%
      add_column(n0 = size[n])
    macrophytes <- rbind(macrophytes, df)

    ### TURBIDITY ###
    df2 <- melt(result$turb, varnames = c("Sim","Time"), value.name = "Tb") %>%
      group_by(Time) %>%
      summarise(mean_T = mean(Tb), se = sd(Tb)/length(Tb)) %>%
      add_column(n0 = size[n])
    turbidity <- rbind(turbidity,df2)
  }
}
```



```

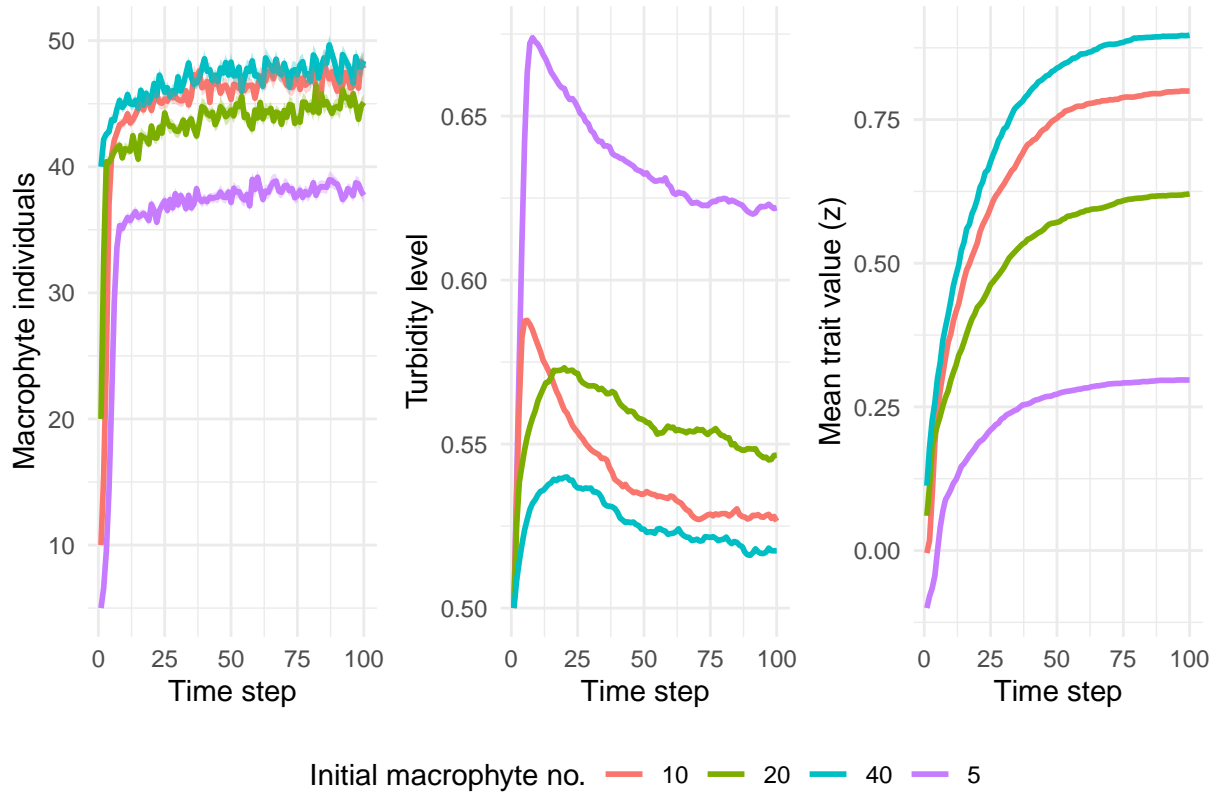
    ### TRAIT MEAN ###
    df3 <- melt(result$Z_means, varnames = c("Sim","Time"), value.name = "zbar")%>%
      group_by(Time) %>%
      summarise(mean_z = mean(zbar), se = sd(zbar)/length(zbar))%>%
      add_column(n0 = size[n])
    trait_val <- rbind(trait_val,df3)
  }

  p1 <- ggplot() +
    geom_line(data = macrophytes, aes(x = Time, y = mean_M, group = n0,
                                      colour = as.character(n0)), lwd = 1) +
    geom_ribbon(data = macrophytes, aes(x = Time, ymin = mean_M-se, ymax = mean_M+se,
                                       group = n0, fill = as.character(n0)),
              alpha = 0.3, show.legend = FALSE)+
    theme_minimal() + labs(x = "Time step", y = "Macrophyte individuals", colour = "Initial macrophyte no.")+
    theme(legend.position = "bottom")
  p2 <- ggplot() +
    geom_line(data = turbidity, aes(x = Time, y = mean_T, group = n0,
                                    colour = as.character(n0)), lwd = 1) +
    geom_ribbon(data = turbidity, aes(x = Time, ymin = mean_T-se, ymax = mean_T+se,
                                     group = n0, fill = as.character(n0)),
              alpha = 0.3, show.legend = FALSE)+
    theme_minimal()+ labs(x = "Time step", y = "Turbidity level", colour = "Initial macrophyte no.")+
    theme(legend.position = "bottom")
  p3 <- ggplot() +
    geom_line(data = trait_val, aes(x = Time, y = mean_z, group = n0,
                                    colour = as.character(n0)), lwd = 1) +
    geom_ribbon(data = trait_val, aes(x = Time, ymin = mean_z-se, ymax = mean_z+se,
                                     group = n0, fill = as.character(n0)),
              alpha = 0.3, show.legend = FALSE)+
    theme_minimal()+ labs(x = "Time step", y = "Mean trait value (z)", colour = "Initial macrophyte no.")+
    theme(legend.position = "bottom")

  ### use patchwork to return ###
  p <- (p1 | p2 | p3) + plot_layout (guides = "collect") &
  theme(legend.position='bottom')
  return(p)
}

vary_initial_mac()

```



```
##### simulate final population size over range of background turbidity #####
##### WITH trait variation #####
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)
}

##### simulate final population size over range of background turbidity #####
##### WITHOUT trait variation #####
M2 <- rep(0,7)
final_mac2 <- 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,
```

```

      zmin = 100, zmax = 100)

#average final macrophyte population size
M2[t-1] <- Macrophyte_sim_summary(result, "stats")[[1]]

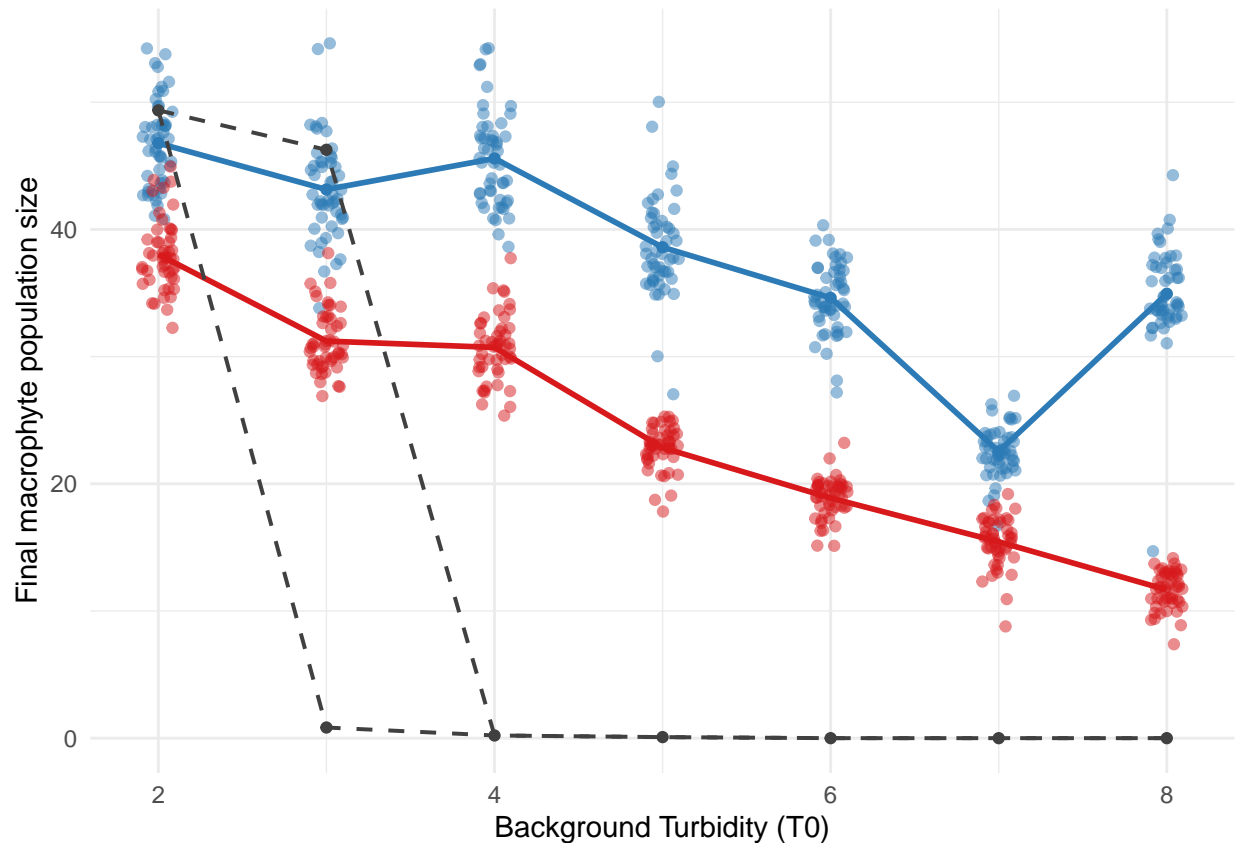
#final macrophyte population size for each simulation
Mac <- Macrophyte_sim_summary(result, summary = "data")[[1]]
Mac <- cbind(T0 = rep(t,10), final_M = Mac[Mac$Time==100,"M"])
#store final size with T0 level
final_mac2 <- rbind(final_mac2, Mac)

}

##### final population values expected from analytic solutions #####
analytic_M <- read.csv("analytic_M.csv")

## plot ##
ggplot()+geom_jitter(data = final_mac, aes(x = T0, y = final_M),
                     col = "#2c7bb6", alpha = 0.5, width = 0.1) +
  geom_line(aes(x = 2:8, y = M), col = "#2c7bb6", lwd = 1) +
  geom_point(aes(x = 2:8, y = M), col = "#2c7bb6") +
  labs(x = "Background Turbidity (T0)", y="Final macrophyte population size")+
  geom_line(aes(x = 2:8, y = M2), col = "#d7191c", lwd = 1) +
  geom_jitter(data = final_mac2, aes(x = T0, y = final_M),
              col = "#d7191c", alpha = 0.5, width = 0.1)+
  geom_point(data = analytic_M, aes(x = T0, y = M), col = "#404040")+
  geom_line(data = analytic_M[-2,], aes(x = T0, y = M), col = "#404040",
            lwd = 0.7, lty = 2) +
  geom_line(data = analytic_M[-3,], aes(x = T0, y = M), col = "#404040",
            lwd = 0.7, lty = 2) +
  theme_minimal()

```



```
## plotting Z values for different initial trait values
result <- runMacrophyte(30, initurb = 0.5, T0 = 2, n0 = 5)
zdata <- Macrophyte_sim_summary(result, summary = "data")[[3]]
z_summary <- zdata %>%
  group_by(Time) %>%
  summarise(mean_z = mean(zbar), se_z = sd(zbar))

result <- runMacrophyte(30, initurb = 0.5, T0 = 2, n0 = 20)
zdata2 <- Macrophyte_sim_summary(result, summary = "data")[[3]]
z_summary2 <- zdata2 %>%
  group_by(Time) %>%
  summarise(mean_z = mean(zbar), se_z = sd(zbar))

result <- runMacrophyte(30, initurb = 0.5, T0 = 2, n0 = 40)
zdata3 <- Macrophyte_sim_summary(result, summary = "data")[[3]]
z_summary3 <- zdata3 %>%
  group_by(Time) %>%
  summarise(mean_z = mean(zbar), se_z = sd(zbar)/sqrt(length(zbar)))

ggplot() + geom_line(data = z_summary, aes(x = Time, y = mean_z), col = "blue") +
  geom_line(data = z_summary2, aes(x = Time, y = mean_z), col = "green") +
  geom_line(data = z_summary3, aes(x = Time, y = mean_z), col = "purple") +
  geom_ribbon(data = z_summary, aes(x = Time, ymin = mean_z-se_z, ymax = mean_z+se_z), fill = "blue", alpha = 0.3) +
  geom_ribbon(data = z_summary2, aes(x = Time, ymin = mean_z-se_z, ymax = mean_z+se_z), fill = "green", alpha = 0.3) +
  geom_ribbon(data = z_summary3, aes(x = Time, ymin = mean_z-se_z, ymax = mean_z+se_z), fill = "purple", alpha = 0.3)
```

Conclusion

Trait variation and evolution increase the resilience of macrophyte population. Consequently, trait variation is extremely important for maintaining shallow lake ecosystems. Furthermore, initial turbidity has no effect on macrophyte population trajectories. This information is important for restoration efforts - reintroduction of macrophytes should be possible at various turbidity levels.