microstegium_elymus_model_summary

Amy Kendig

12/5/2019

Summary

Simulation of *Microstegium vimineum* and *Elymus virgincus* over time. *M. vimineum* is an invasive annual grass and *E. virginicus* is a native perennial grass. *E. virginicus* grows by itself for 100 years, and then *M. vimineum* is introduced. Disease effects arise 100 years later. Population dynamics are similar in the absence of disease and when both species are susceptible to disese. In both cases, *M. vimineum* reduces the size of the *E. virginicus* population. *M. vimineum* is able to recover population losses due to disease. *E. virginicus* seedlings are impacted more than adults. When only *M. vimineum* is susceptible to disease, its population crashes.

Set-up

```r
# clear all existing data
rm(list = ls())

# open libraries
library(data.table)
library(plotly)
library(cowplot)
library(popbio)
library(tidyverse)

# plotting parameters
axisText = 10
axisTitle = 12
legendText = 10
legendTitle = 0

# import data
params <- read_csv("../data/mv_ev_model_parameters_120319.csv", col_types = cols())
```

Parameters

Formatting:

- subscripts follow "."
- A = annual plant
- L = annual litter
- P = perennial adult (at least 1 year old)
- S = perennial seedling (germinated that spring)

- Composite parameters are capitalized

Settings:

```
# simulation time
simtime = 500

# invasion time
invtime = 100

# disease time
distime = invtime + 100

# initial conditions
N0.A = 0 # initial annual population size
N0.S = 1 # initial perennial seedling population size
N0.P = 0 # initial perennial adult population size
L0 = 0 # initial annual litter amount
Ni.A = 1 # introduction of annual
```

Literature:

parameter	symbol	value	units	reference
annual germination without litter	g.A	0.8000	year^{-1}	Redwood et al. 2018
litter suppression of germination	beta	0.0065	g^{-1} year^{-1}	Foster and Gross 1998
annual seed survival	s.A	0.1300	year^{-1}	Redwood et al. 2018
annual biomass without competition	v.A	17.4000	g year^{-1}	Wilson et al. 2015
annual biomass-seed conversion	y.A	86.0000	seeds g^{-1} year^{-1}	Wilson et al. 2015, Redwood et al. 2018
annual litter-biomass conversion	a	0.9500	year^{-1}	DeMeester and Richter 2010
litter decomposition rate	b	0.6700	year^{-1}	DeMeester and Richter 2010
perennial germination without litter	g.S	0.8000	year^{-1}	Garrison and Stier 2010
perennial seed survival	s.S	1.0000	year^{-1}	Garrison and Stier 2010
adult perennial seed production without competition	f.P	22.0000	seeds year^{-1}	Stevens 1957, Garrison and Stier 2010
seedling perennial seed production without competition	f.1	2.0000	seeds year^{-1}	1/10 of f.P
perennial seedling survival	s.1	0.4000	year^{-1}	Mottl et al. 2006
perennial adult survival	s.P	0.8900	year^{-1}	Mottl et al. 2006
competitive effect of annual	alpha.A	0.0200	year^{-1}	average of alpha.iA
competitive effect of perennial seedling	alpha.S	0.0020	year^{-1}	average of alpha.iS
competitive effect of perennial adult	alpha.P	0.0200	year^{-1}	average of alpha.iP
reduction in biomass due to disease	dis	0.3500	year^{-1}	Flory et al. 2011, Stricker et al. 2016

Model

Population equations:

Assume counts are being conducted in the spring

N.A[t+1] = s.A * (1-G.A) * N.A[t] + G.A * V.A * y.A * N.A[t]
annual seeds = seed bank survival + seed production

L[t+1] = a * G.A * V.A * N.A[t] + (1-b) * L[t] annual litter = biomass from previous fall + decomposition

N.S[t+1] = s.S * (1-G.S) * N.S[t] + G.S * F.1 * N.S[t] + F.P * N.P[t]
perennial seeds = seed bank survival + seedling seed production + adult seed production

N.P[t+1] = s.P * N.P[t] + s.1 * G.S * N.S[t]
perennial adults = survival + seedling maturation

Litter suppression of germination

G.A = g.A / (1 + beta * L[t])
annual seed germination = germination in the absence of litter / reduction due to litter

G.S = g.S / (1 + beta * L[t])
perennial seed germination = germination in the absence of litter / reduction due to litter

Density-dependence on growth and fecundity

V.A = v.A / (1 + alpha.A * G.A * N.A[t] + alpha.P * N.P[t] + alpha.S * G.S * N.S[t]) annual growth = growth in the absence of competition / (competition from annuals + competition from perennial adults + competition from perennial seedlings)

F.1 = f.1 / (1 + alpha.A * G.A * N.A[t] + alpha.P * N.P[t] + alpha.S * G.S * N.S[t]) perennial seedling seed production = seed production in the absence of competition / (competition from annuals + competition from perennial adults + competition from perennial seedlings)

F.P = f.P / (1 + alpha.A * G.A * N.A[t] + alpha.P * N.P[t] + alpha.S * G.S * N.S[t]) perennial adult seed production = seed production in the absence of competition / (competition from annuals + competition from perennial adults + competition from perennial seedlings)

Function

```r
simFun = function(params, N0.A, N0.S, N0.P, L0, Ni.A, simtime, invtime,
    distime) {

    # define parameters
    g.A = filter(params, symbol == "g.A")$value
    beta = filter(params, symbol == "beta")$value
    s.A = filter(params, symbol == "s.A")$value
    v.A = filter(params, symbol == "v.A")$value
    y.A = filter(params, symbol == "y.A")$value
    a = filter(params, symbol == "a")$value
    b = filter(params, symbol == "b")$value
    g.S = filter(params, symbol == "g.S")$value
    s.S = filter(params, symbol == "s.S")$value
    f.P = filter(params, symbol == "f.P")$value
    f.1 = filter(params, symbol == "f.1")$value
    s.1 = filter(params, symbol == "s.1")$value
    s.P = filter(params, symbol == "s.P")$value
    alpha.A = filter(params, symbol == "alpha.A")$value
    alpha.S = filter(params, symbol == "alpha.S")$value
    alpha.P = filter(params, symbol == "alpha.P")$value
    dis = filter(params, symbol == "dis")$value

    # initialize populations
    N.A = rep(NA, simtime)
    N.S = rep(NA, simtime)
```

```r
    N.P = rep(NA, simtime)
    L = rep(NA, simtime)

    N.A[1] = N0.A
    N.S[1] = N0.S
    N.P[1] = N0.P
    L[1] = L0

    # simulate population dynamics
    for (t in 1:(simtime - 1)) {

        # introduce annual at t=100
        N.A[t] = ifelse(N0.A == 0 & t == invtime, Ni.A, N.A[t])

        # calulate parameters to introduce disease at distime
        growth.A = ifelse(t < distime, v.A, dis * g.A)

        # reduce germination due to litter
        G.A = g.A/(1 + beta * L[t])
        G.S = g.S/(1 + beta * L[t])

        # reduce fecundity or growth due to competition
        V.A = growth.A/(1 + alpha.A * G.A * N.A[t] + alpha.P * N.P[t] +
            alpha.S * G.S * N.S[t])
        F.1 = f.1/(1 + alpha.A * G.A * N.A[t] + alpha.P * N.P[t] + alpha.S *
            G.S * N.S[t])
        F.P = f.P/(1 + alpha.A * G.A * N.A[t] + alpha.P * N.P[t] + alpha.S *
            G.S * N.S[t])

        # population size
        N.A[t + 1] = s.A * (1 - G.A) * N.A[t] + G.A * V.A * y.A * N.A[t]
        L[t + 1] = a * G.A * V.A * N.A[t] + (1 - b) * L[t]
        N.S[t + 1] = s.S * (1 - G.S) * N.S[t] + G.S * F.1 * N.S[t] + F.P *
            N.P[t]
        N.P[t + 1] = s.P * N.P[t] + s.1 * G.S * N.S[t]

        # correct to prevent negative numbers
        N.A[t + 1] = ifelse(N.A[t + 1] < 1, 0, N.A[t + 1])
        L[t + 1] = ifelse(L[t + 1] < 1, 0, L[t + 1])
        N.S[t + 1] = ifelse(N.S[t + 1] < 1, 0, N.S[t + 1])
        N.P[t + 1] = ifelse(N.P[t + 1] < 1, 0, N.P[t + 1])
    }

    # save data
    dfN = data.frame(time = rep(1:simtime, 4), N = c(N.S, N.P, N.A, L),
        species = rep(c("Elymus seedling", "Elymus adult", "Microstegium",
            "Microstegium litter"), each = simtime))

    # return
    return(dfN)
}
```
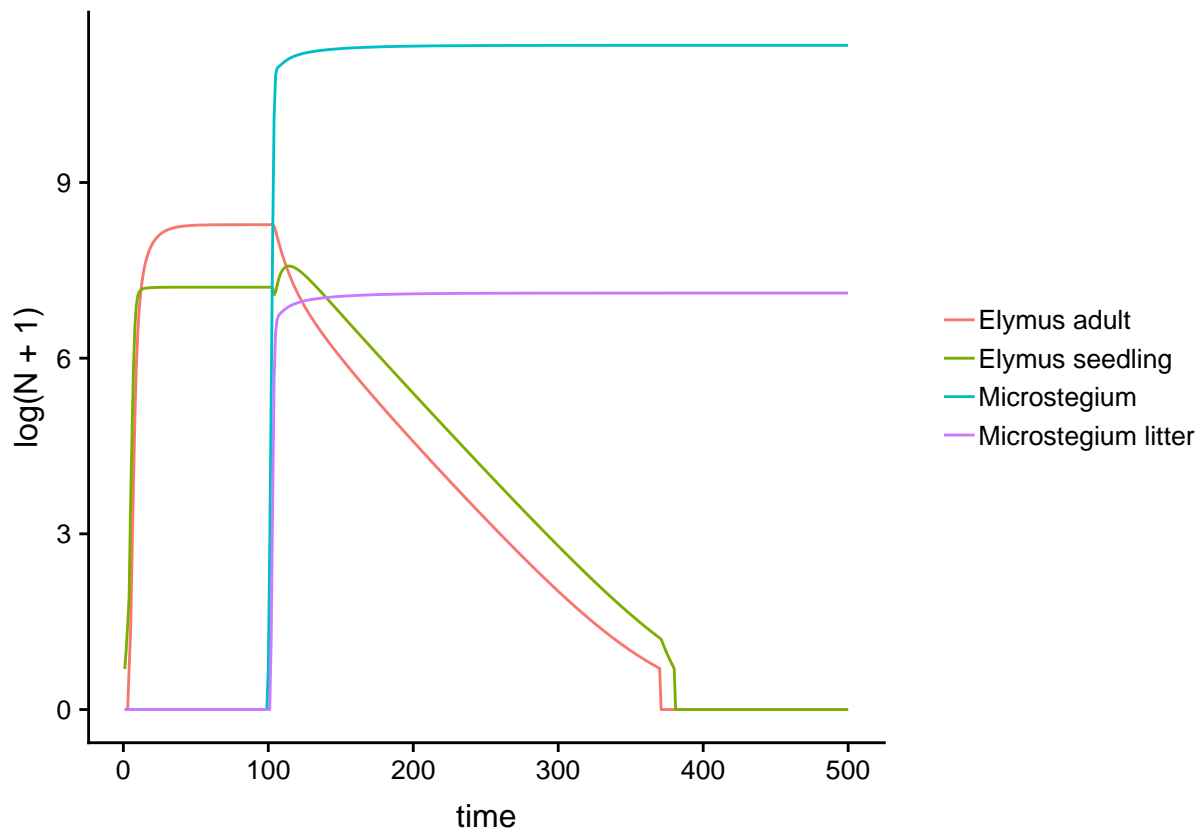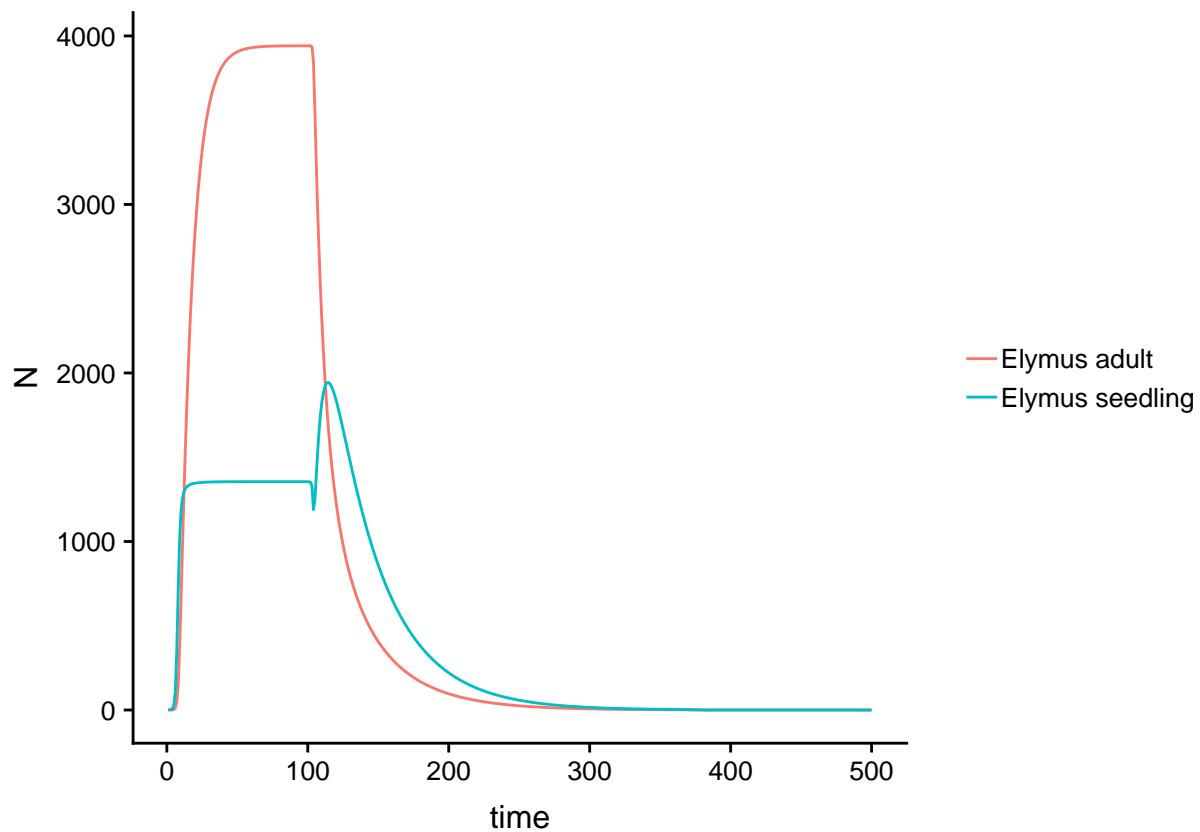
Simulations

No disease

```
nodisSim <- simFun(params, N0.A, N0.S, N0.P, L0, Ni.A, simtime, invtime, distime = simtime + 10)
nodisSim %>%
  ggplot(aes(x = time, y = log(N+1), color = species)) +
  geom_line() +
  theme(axis.text = element_text(size = axisText),
        axis.title = element_text(size = axisTitle),
        legend.text = element_text(size = legendText),
        legend.title = element_text(size = legendTitle))
```



```
nodisSim %>%
  filter(species %in% c("Elymus adult", "Elymus seedling")) %>%
  ggplot(aes(x = time, y = N, color = species)) +
  geom_line() +
  theme(axis.text = element_text(size = axisText),
        axis.title = element_text(size = axisTitle),
        legend.text = element_text(size = legendText),
        legend.title = element_text(size = legendTitle))
```

Add disease

```
disSim <- simFun(params, N0.A, N0.S, N0.P, L0, Ni.A, simtime, invtime, distime)
disSim %>%
  ggplot(aes(x = time, y = log(N+1), color = species)) +
  geom_line() +
  theme(axis.text = element_text(size = axisText),
        axis.title = element_text(size = axisTitle),
        legend.text = element_text(size = legendText),
        legend.title = element_text(size = legendTitle))
```
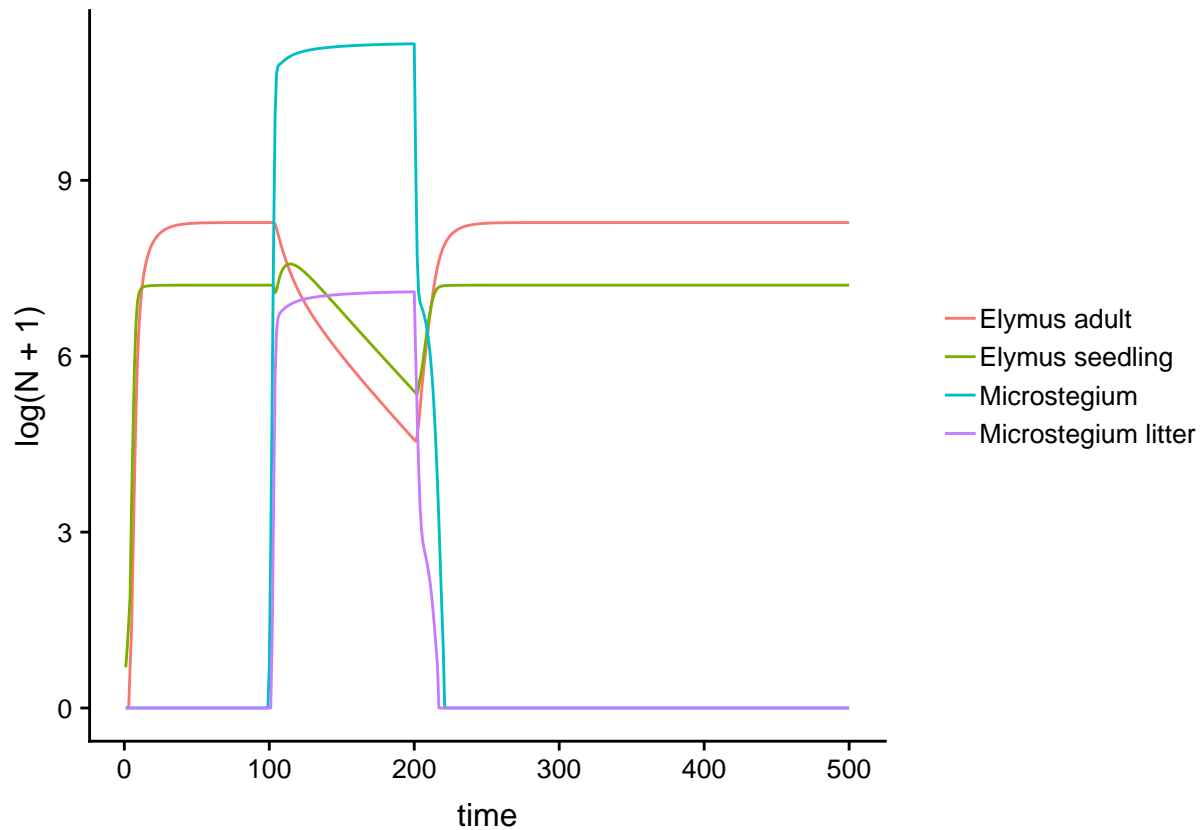
Time to extinction?

```
nodisSim %>%
  filter(species == "Elymus seedling" & N == 0) %>%
  filter(time == min(time))
```

```
##   time N         species
## 1  381 0 Elymus seedling
```

```
disSim %>%
  filter(species == "Microstegium" & N == 0 & time > invtime) %>%
  filter(time == min(time))
```

```
##   time N      species
## 1  221 0 Microstegium
```