microstegium-elymus-model-summary

Amy Kendig

10/17/2019

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_101519.csv", col_types = cols())
aparams <- read_csv("../data/mv_ev_model_alt_parameters_101519.csv", col_types = cols())
```

Parameters

Formatting:

- subscripts follow "."
- p = perennial adult (at least 1 year old)
- s = perennial seedling (germinated that spring)
- a = annual
- L = annual litter

Edit:

```
# derive related parameters
dparams <- tibble(
  parameter = c("annual-adult competition",
                "seedling-adult competition",
                "adult-annual competition",
                "adult-seedling competition",
                "adult-adult competition",
                "seedling seed production"),
  symbol = c("alpha.pa",
             "alpha.ps",
             "alpha.ap",
             "alpha.sp",
             "alpha.pp",
             "lambda.s"),
  value = c(filter(params, symbol == "alpha.sa")$value/10,
            filter(params, symbol == "alpha.sa")$value/10,
            filter(params, symbol == "alpha.as")$value*10,
            filter(params, symbol == "alpha.as")$value*10,
            filter(params, symbol == "alpha.as")$value*10,
            filter(params, symbol == "lambda.p")$value/10)
) %>%
  mutate(units = c(rep("year^-1^", 5), "seeds year^-1^"),
         reference = "derived")

# merge with main parameters
params2 <- full_join(params,
                     dparams,
                     by = c("parameter", "symbol", "value", "units", "reference"))

# simulation time
years = 500
```

Table 1: Model parameter values

parameter	symbol	value	units	reference
perennial adult survival	m.p	0.9500	year^{-1}	Malmstrom et al. 2005
annual seed survival	s.a	0.7400	year^{-1}	Huebner 2011
perennial seed survival	s.s	0.7600	year^{-1}	Robocker et al. 1953
annual germination	gamma.a	0.7000	year^{-1}	Warren et al. 2013
perennial germination	gamma.s	0.6600	year^{-1}	Robocker et al. 1953
litter suppression of annual germination	alpha.aL	-0.0009	g^{-1} year^{-1}	Foster and Gross 1998
litter suppression of perennial germination	alpha.sL	-0.0009	g^{-1} year^{-1}	Foster and Gross 1998
litter decomposition rate	b	0.4400	year^{-1}	Kourtev et al. 2002
annual summer survival	h.a	0.9500	year^{-1}	Warren et al. 2013
perennial seedlingsummer survival	h.s	0.4000	year^{-1}	Mottl et al. 2006
perennial adult summer survival	h.p	0.8300	year^{-1}	Mottl et al. 2006
annual seed production	lambda.a	6500.0000	seeds year^{-1}	Wilson et al. 2015
perennial seed production	lambda.p	435.0000	seeds year^{-1}	Stevens 1957
annual-annual competition	alpha.aa	0.1220	year^{-1}	Leicht et al. 2005
seedling-annual competition	alpha.as	0.3570	year^{-1}	Leicht et al. 2005
seedling-seedling competition	alpha.ss	0.0020	year^{-1}	Leicht et al. 2005
annual-seedling competition	alpha.sa	0.7240	year^{-1}	Leicht et al. 2005
biomass-seed conversion	c.a	0.0050	g seeds^{-1} year^{-1}	Wilson et al. 2015

parameter	symbol	value	units	reference
disease suppression of annual seed production	tol.a	0.1900	year^{-1}	Flory et al. 2011
disease suppression of perennial seed production	tol.p	0.1900	year^{-1}	Flory et al. 2011
annual-adult competition	alpha.pa	0.0724	year^{-1}	derived
seedling-adult competition	alpha.ps	0.0724	year^{-1}	derived
adult-annual competition	alpha.ap	3.5700	year^{-1}	derived
adult-seedling competition	alpha.sp	3.5700	year^{-1}	derived
adult-adult competition	alpha.pp	3.5700	year^{-1}	derived
seedling seed production	lambda.s	43.5000	seeds year^{-1}	derived

Table 2: Alternative model parameter values

parameter	symbol	value	units	reference
annual germination	gamma.a	0.29	year^{-1}	Huebner 2011
annual intraspecific seed competition	alpha.aa	0.001, 0.015, 0.054	year^{-1}	Leicht et al. 2005
annual interspecific competition	alpha.as	0.054, 0.910, 17.919	year^{-1}	Leicht et al. 2005
perennial intraspecific competition	alpha.ss	0.006, 0.011, 0.049	year^{-1}	Leicht et al. 2005
perennial interspecific seed competition	alpha.sa	1.347, 9.574, 23.070	year^{-1}	Leicht et al. 2005
disease suppression of seed production	tol	0.6	year^{-1}	Stricker et al. 2016

Model

Population equations:

Assume counts are being conducted in the fall

N.s[t+1] = s.s * (1-g.s) * N.s[t] + g.s * h.s * f.s * N.s[t] + m.p * f.p * N.p[t]
perennial seeds = seed bank survival + seedling seed production + adult seed production

N.p[t+1] = m.p * N.p[t] + g.s * h. s * N.s[t]
perennial adults = survival + seedling maturation

N.a[t+1] = s.a * (1-g.a) * N.a[t] + g.a * h.a * f.a * N.a[t]
annual seeds = seed bank survival + seed production

L[t+1] = c.a * g.a * h.a * N.a[t] + (1-b) * L[t]
annual litter = biomass from previous fall + decomposition

Density-dependence on fecundity

f.s = lam.s / (1 + alpha.ss * g.s * h.s * N.s[t] + alpha.sp * m.p * N.p[t] + alpha.sa * g.a * h.a * N.a[t])
perennial seedling fecundity = fecundity in the absence of competition / (perennial seedling competition + perennial adult competition + annual competition)

f.p = lam.p / (1 + alpha.ps * g.s * h.s * N.s[t] + alpha.pp * m.p * N.p[t] + alpha.pa * g.a * h.a * N.a[t])
perennial adult fecundity = fecundity in the absence of competition / (perennial seedling competition + perennial adult competition + annual competition)

f.a = lam.a / (1 + alpha.as * g.s * h.s * N.s[t] + alpha.ap * m.p * N.p[t] + alpha.aa * g.a * h.a * N.a[t])
annual fecundity = fecundity in the absence of competition / (perennial seedling competition + perennial adult competition + annual competition)

Litter suppression

g.s = gamma.s + (alpha.sL * L[t])
perennial seed germination = germination in the absence of litter + reduction due to litter (alpha.sL < 0, g.s constrained to >= 0)

g.a = gamma.a + (alpha.aL * L[t])
annual seed germination = germination in the absence of litter + reduction due to litter (alpha.aL < 0, g.a constrained to >= 0)

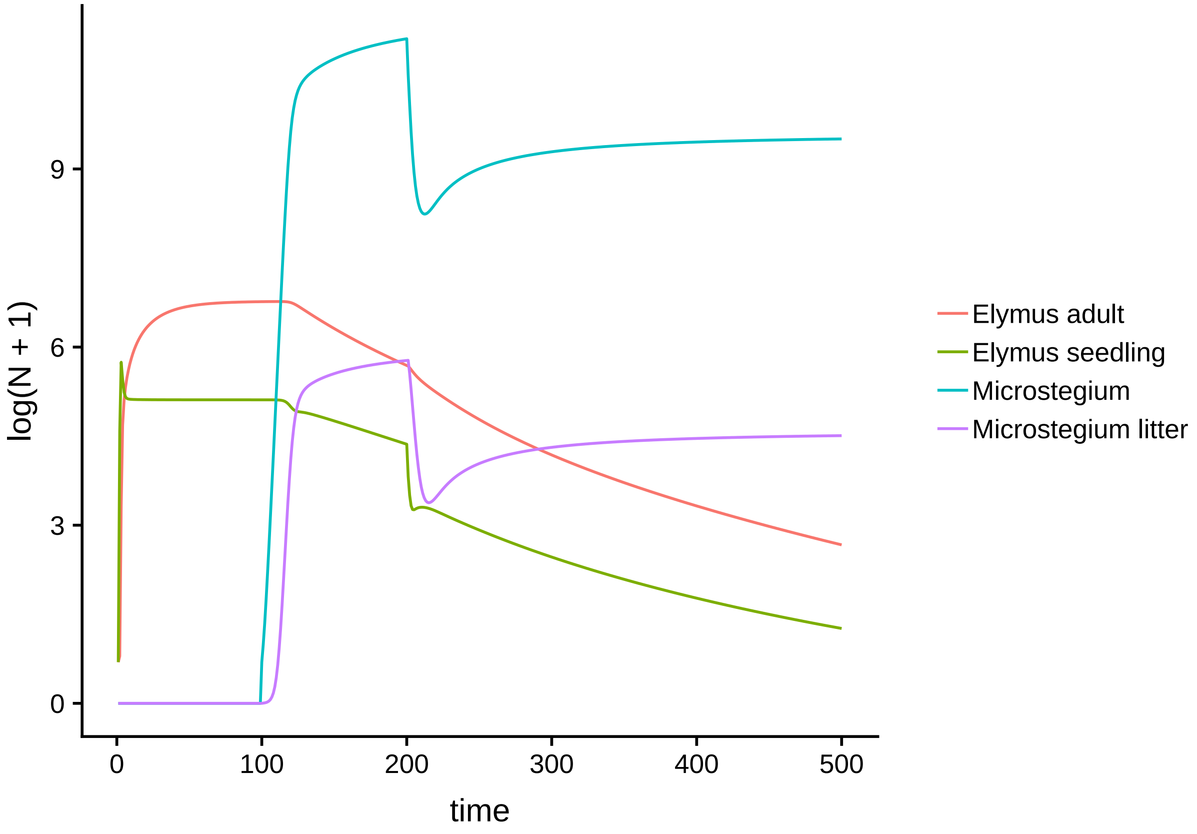
Function

```
simFun = function(params, N0.a, N0.s, N0.p, L0, Ni.a, simtime) {

    # define parameters
    m.p = filter(params, symbol == "m.p")$value
    s.a = filter(params, symbol == "s.a")$value
    s.s = filter(params, symbol == "s.s")$value
    gamma.a = filter(params, symbol == "gamma.a")$value
    gamma.s = filter(params, symbol == "gamma.s")$value
    alpha.aL = filter(params, symbol == "alpha.aL")$value
    alpha.sL = filter(params, symbol == "alpha.sL")$value
    b = filter(params, symbol == "b")$value
    h.a = filter(params, symbol == "h.a")$value
    h.s = filter(params, symbol == "h.s")$value
    h.p = filter(params, symbol == "h.p")$value
    lambda.a = filter(params, symbol == "lambda.a")$value
    lambda.p = filter(params, symbol == "lambda.p")$value
    lambda.s = filter(params, symbol == "lambda.s")$value
    alpha.aa = filter(params, symbol == "alpha.aa")$value
    alpha.as = filter(params, symbol == "alpha.as")$value
    alpha.ap = filter(params, symbol == "alpha.ap")$value
    alpha.sa = filter(params, symbol == "alpha.sa")$value
    alpha.ss = filter(params, symbol == "alpha.ss")$value
    alpha.sp = filter(params, symbol == "alpha.sp")$value
    alpha.pa = filter(params, symbol == "alpha.pa")$value
    alpha.ps = filter(params, symbol == "alpha.ps")$value
    alpha.pp = filter(params, symbol == "alpha.pp")$value
    c.a = filter(params, symbol == "c.a")$value
    tol.a = filter(params, symbol == "tol.a")$value
    tol.p = filter(params, symbol == "tol.p")$value

    # initialize populations
    N.a = rep(NA, simtime)
    N.s = rep(NA, simtime)
    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 == 100, Ni.a, N.a[t])

        # calulate parameters to introduce disease at t=200
        lam.a = ifelse(t < 200, lambda.a, tol.a * lambda.a)
        lam.p = ifelse(t < 200, lambda.p, tol.p * lambda.p)
        lam.s = ifelse(t < 200, lambda.s, tol.p * lambda.s)

        # reduce germination due to litter
        g.s = gamma.s + (alpha.sL * L[t])
        g.s = ifelse(g.s < 0, 0, g.s)
        g.a = gamma.a + (alpha.aL * L[t])
        g.a = ifelse(g.a < 0, 0, g.a)

        # reduce fecundity due to competition
        f.s = lam.s/(1 + alpha.ss * g.s * h.s * N.s[t] + alpha.sp * m.p *
            N.p[t] + alpha.sa * g.a * h.a * N.a[t])
        f.p = lam.p/(1 + alpha.ps * g.s * h.s * N.s[t] + alpha.pp * m.p *
            N.p[t] + alpha.pa * g.a * h.a * N.a[t])
        f.a = lam.a/(1 + alpha.as * g.s * h.s * N.s[t] + alpha.ap * m.p *
            N.p[t] + alpha.aa * g.a * h.a * N.a[t])

        # population size
        N.s[t + 1] = s.s * (1 - g.s) * N.s[t] + g.s * h.s * f.s * N.s[t] +
            m.p * f.p * N.p[t]
        N.p[t + 1] = m.p * N.p[t] + g.s * h.s * N.s[t]
        N.a[t + 1] = s.a * (1 - g.a) * N.a[t] + g.a * h.a * f.a * N.a[t]
        L[t + 1] = c.a * g.a * h.a * N.a[t] + (1 - b) * L[t]

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

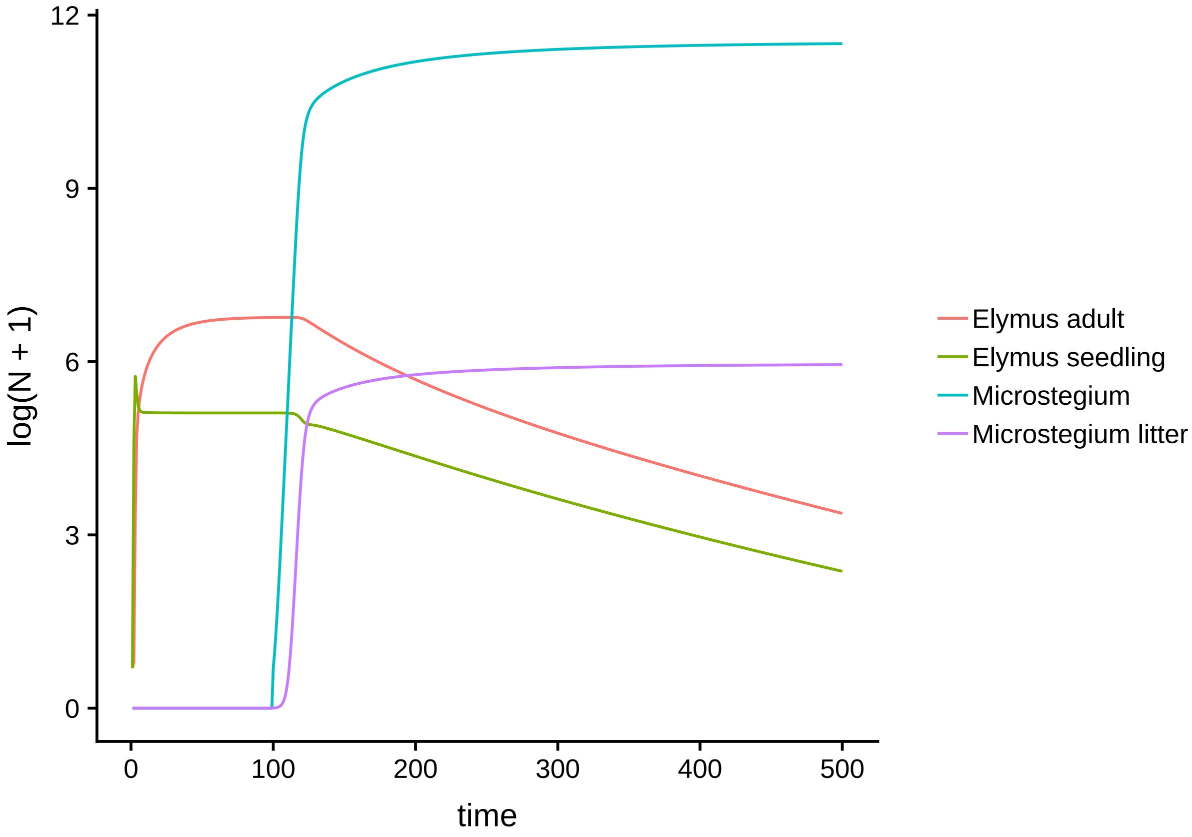
Default parameters

```
simFun(params = params2,
       N0.a = 0,
       N0.s = 1,
       N0.p = 1,
       L0 = 0,
       Ni.a = 1,
       simtime=years) %>%
  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))
```



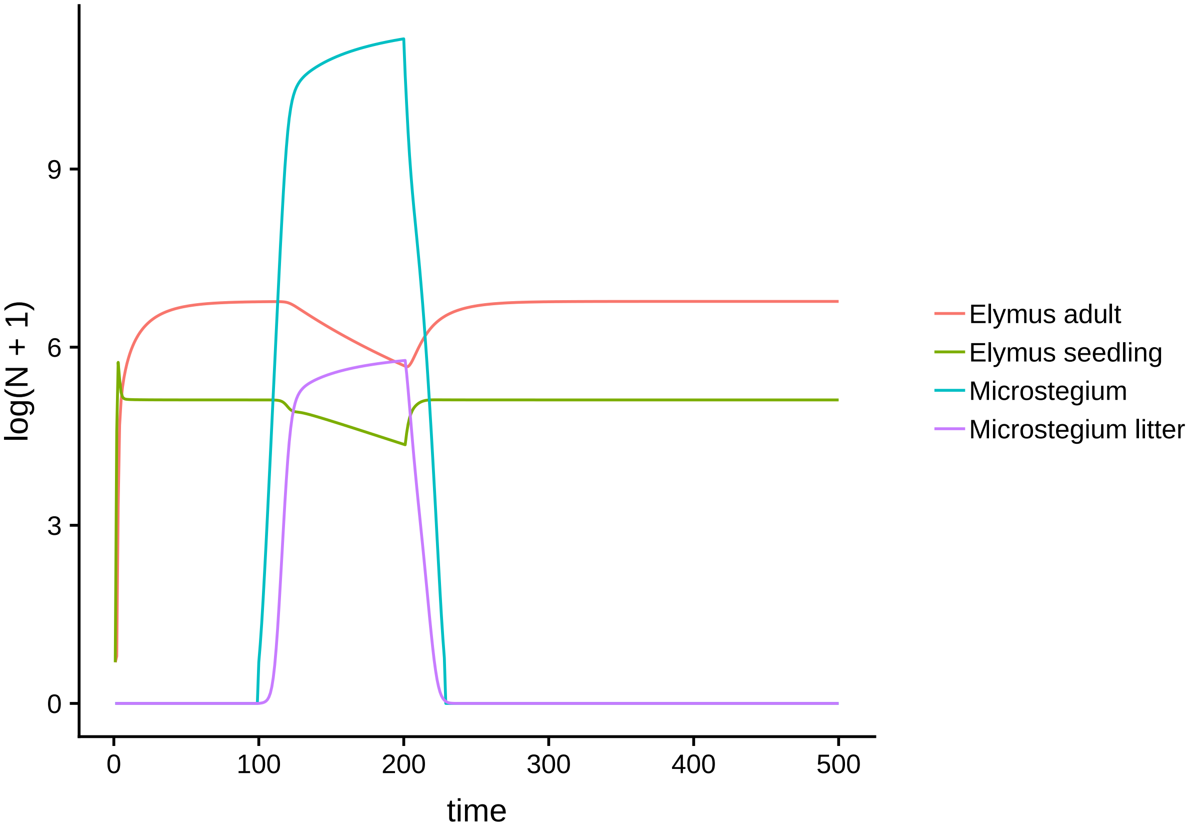
Remove disease

```
simFun(params = params2 %>% mutate(value = case_when(symbol %in% c("tol.a", "tol.p") ~ 1,
                                                     TRUE ~ value)),
       N0.a = 0,
       N0.s = 1,
       N0.p = 1,
       L0 = 0,
       Ni.a = 1,
       simtime=years) %>%
  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))
```



Remove disease from perennial

```
simFun(params = params2 %>% mutate(value = case_when(symbol == "tol.p" ~ 1,
                                                     TRUE ~ value)),
       N0.a = 0,
       N0.s = 1,
       N0.p = 1,
       L0 = 0,
       Ni.a = 1,
       simtime=years) %>%
  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))
```


log(N + 1)
9
6
3
0
0
100
200
300
400
500
time
Elymus adult
Elymus seedling
Microstegium
Microstegium litter