

microstegium_model_kortessis_comparison

Amy Kendig

11/19/2019

Summary

The goal is to compare my model with Nick's, which are the same except that:

- I model decomposition as exponential decay between years instead of proportion leftover (1-d) - changed mine to match
- I made litter suppression linear with a manual cut-off instead of using the Beverton-Holt form - changed mine to match
- I use an establishment rate (i.e., summer survival) instead of a variable representing size (biomass/seed) for both annual and perennial. I don't think the relationship between size and seed production changes with competition. These variables are both 1 below (i.e., no scenario testing).
- I allow for different competitive effects of different species. I set them equal to each other for this (i.e., no scenario testing).
- My transition from seed count to biomass production is different because I convert the individuals to biomass and Nick's model has biomass as a component of the annual seed model (vigor). He also uses an overwinter decomposition rate (my litter build-up is slower because the conversion parameter is so small).
- Scenario 3 (aklitter): large change in density
- I include establishment (for seeds) and winter survival (for adults) in the competition pressure.
- Scenario 4 (akcomp): minor change in density - The perennial model is different because I have first-year seedlings producing seeds and experiencing competition. They have the option to cause competitive effects, but I set that to 0.
- Scenario 5 (akper): no change in density
- I account for perennial adult survival before calculating seed production.
- Scenario 6 (akseed): large change in density.
- A change to make to both models? The per-seed contribution to litter should change with competition (and potentially disease).

Set-up

```
# 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 parameters
params <- read_csv("../data/mv_ev_kortessis_comparison_111919.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

Settings:

```
# initial conditions
NO.a = 5
NO.s = 5
NO.p = 5
LO = 2

# simulation time
years = 300
```

Table 1: Model parameter values

parameter	symbol	value	units
perennial adult survival	m.p	0.800	year ⁻¹
annual seed survival	s.a	0.800	year ⁻¹
perennial seed survival	s.s	0.500	year ⁻¹
annual germination	gamma.a	1.000	year ⁻¹
perennial germination	gamma.s	0.500	year ⁻¹
litter suppression of annual germination	alpha.aL	1.500	g ⁻¹ year ⁻¹
litter suppression of perennial germination	alpha.sL	0.250	g ⁻¹ year ⁻¹
litter decomposition rate	b	0.200	year ⁻¹
annual summer survival	h.a	1.000	year ⁻¹
perennial seedlingsummer survival	h.s	1.000	year ⁻¹
perennial adult summer survival	h.p	1.000	year ⁻¹
annual seeds per biomass	v.a	1.000	year ⁻¹
perennial seeds per biomass	v.p	1.000	year ⁻¹
annual seed production	lambda.a	13.500	seeds year ⁻¹
perennial seed production	lambda.p	3.000	seeds year ⁻¹
seedling seed production	lambda.s	0.300	seeds year ⁻¹
annual-annual competition	alpha.aa	0.250	year ⁻¹
seedling-annual competition	alpha.as	0.000	year ⁻¹
perennial-annual competition	alpha.ap	0.250	year ⁻¹
seedling-seedling competition	alpha.ss	0.000	year ⁻¹
annual-seedling competition	alpha.sa	2.500	year ⁻¹
perennial-seedling competition	alpha.sp	2.500	year ⁻¹
perennial-perennial competition	alpha.pp	0.250	year ⁻¹
annual-perennial competition	alpha.pa	0.250	year ⁻¹
seedling-perennial competition	alpha.ps	0.000	year ⁻¹

parameter	symbol	value	units
annual biomass per seed	c.a	0.005	g seeds ⁻¹ year ⁻¹
overwinter litter dcomposition	w.a	0.500	year ⁻¹

Amy's model description

Population equations:

$$N.a[t+1] = s.a * (1-g.a[t]) * N.a[t] + g.a[t] * h.a * f.a[t] * N.a[t]$$

annual seeds = seed bank survival + seed production

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

Assume counts are being conducted in the fall

$$N.s[t+1] = s.s * (1-g.s[t]) * N.s[t] + g.s[t] * h.s * f.s[t] * N.s[t] + m.p * f.p[t] * N.p[t]$$

perennial seeds = seed bank survival + seedling seed production + adult seed production

$$N.p[t+1] = m.p * N.p[t] + g.s[t] * h.s * N.s[t]$$

perennial adults = survival + seedling maturation

Density-dependence on fecundity:

$$f.s[t] = \text{lambda.s} / (1 + \alpha.ss * g.s[t] * h.s * N.s[t] + \alpha.sp * m.p * N.p[t] + \alpha.sa * g.a[t] * h.a * N.a[t])$$

perennial seedling fecundity = fecundity in the absence of competition / (perennial seedling competition + perennial adult competition + annual competition)

$$f.p[t] = \text{lambda.p} / (1 + \alpha.ps * g.s[t] * h.s * N.s[t] + \alpha.pp * m.p * N.p[t] + \alpha.pa * g.a[t] * h.a * N.a[t])$$

perennial adult fecundity = fecundity in the absence of competition / (perennial seedling competition + perennial adult competition + annual competition)

$$f.a[t] = \text{lambda.a} / (1 + \alpha.as * g.s[t] * h.s * N.s[t] + \alpha.ap * m.p * N.p[t] + \alpha.aa * g.a[t] * h.a * N.a[t])$$

annual fecundity = fecundity in the absence of competition / (perennial seedling competition + perennial adult competition + annual competition)

Litter suppression:

$$g.a[t] = \text{gamma.a} / (1 + \alpha.aL * L[t])$$

annual seed germination = germination in the absence of litter / reduction due to litter

$$g.s[t] = \text{gamma.s} / (1 + \alpha.sL * L[t])$$

perennial seed germination = germination in the absence of litter / reduction due to litter

Amy's model function

```
simFun_a = function(params, NO.a, NO.s, NO.p, L0, simtime, alt.comp, alt.per,
  alt.seed) {

  # 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
v.a = filter(params, symbol == "v.a")$value
v.p = filter(params, symbol == "v.p")$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
w.a = filter(params, symbol == "w.a")$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)) {

  # reduce germination due to litter
  g.a = gamma.a/(1 + alpha.aL * L[t])
  g.s = gamma.s/(1 + alpha.sL * L[t])

  # reduce fecundity due to competition (include survival)
  f.a1 = lambda.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])
  f.s1 = lambda.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.p1 = lambda.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])

  # reduce fecundity due to competition (don't include survival)

```

```

f.a2 = lambda.a/(1 + alpha.as * g.s * N.s[t] + alpha.ap * N.p[t] +
  alpha.aa * g.a * N.a[t])
f.s2 = lambda.s/(1 + alpha.ss * g.s * N.s[t] + alpha.sp * N.p[t] +
  alpha.sa * g.a * N.a[t])
f.p2 = lambda.p/(1 + alpha.ps * g.s * N.s[t] + alpha.pp * N.p[t] +
  alpha.pa * g.a * N.a[t])

# choose fecundity
f.a = ifelse(alt.comp == T, f.a2, f.a1)
f.s = ifelse(alt.comp == T, f.s2, f.s1)
f.p = ifelse(alt.comp == T, f.p2, f.p1)

# remove first-year seed production
f.s = ifelse(alt.per == T, 0, f.s)

# choose adult seed production
seed.p = ifelse(alt.seed == T, f.p, m.p * f.p)

# population size
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]
N.s[t + 1] = s.s * (1 - g.s) * N.s[t] + g.s * h.s * f.s * N.s[t] +
  seed.p * N.p[t]
N.p[t + 1] = m.p * N.p[t] + g.s * h.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] < 0, 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])
}

# total perennial population
N.t = N.s + N.p

# save data
dfN = data.frame(time = rep(1:simtime, 5), N = c(N.a, L, N.s, N.p,
  N.t), species = rep(c("Annual", "Annual litter", "Perennial seedling",
  "Perennial adult", "Perennial"), each = simtime))

# return
return(dfN)
}

```

Nick's model function

```

simFun_n = function(params, NO.a, NO.s, NO.p, LO, 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")$value
v.a = filter(params, symbol == "v.a")$value
v.p = filter(params, symbol == "v.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
w.a = filter(params, symbol == "w.a")$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)) {

  # reduce germination due to litter
  g.a = gamma.a/(1 + alpha.aL * L[t])
  g.s = gamma.s/(1 + alpha.sL * L[t])

  # reduce fecundity due to competition
  f.a = lambda.a/(1 + alpha.ap * N.p[t] + alpha.aa * g.a * N.a[t])
  f.p = lambda.p/(1 + alpha.pp * N.p[t] + alpha.pa * g.a * N.a[t])

  # population size
  N.a[t + 1] = s.a * (1 - g.a) * N.a[t] + g.a * v.a * f.a * N.a[t]
  L[t + 1] = w.a * (g.a * v.a * N.a[t] + (1 - b) * L[t])
  N.s[t + 1] = s.s * (1 - g.s) * N.s[t] + v.p * f.p * N.p[t]
  N.p[t + 1] = m.p * N.p[t] + 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] < 0, 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])
  }

  # total perennial population
  N.t = N.s + N.p

  # save data
  dfN = data.frame(time = rep(1:simtime, 5), N = c(N.a, L, N.s, N.p,
    N.t), species = rep(c("Annual", "Annual litter", "Perennial seedling",
    "Perennial adult", "Perennial"), each = simtime))

  # return
  return(dfN)
}

```

Simulations

Function for creating figures:

```

figfun <- function(moda, modn, filename){

  # Format data
  mod <- moda %>%
  mutate(model = "Amy's") %>%
  full_join(modn %>% mutate(model = "Nick's"), by = c("time", "N", "species", "model")) %>%
  mutate(life_history = case_when(species %in% c("Annual", "Annual litter") ~ "annual",
    TRUE ~ "perennial"),
    status = case_when(species %in% c("Annual", "Perennial", "Perennial adult") ~ "annual/adults",
    TRUE ~ "litter/seedlings"),
    model = factor(model, levels = c("Nick's", "Amy's")))

  # Total population figure
  tot_fig <- mod %>%
  filter(species %in% c("Annual", "Perennial")) %>%
  ggplot(aes(x = time, y = N, color = life_history)) +
  geom_line(size = 1.5) +
  facet_wrap(~model) +
  xlab("Time") +
  ylab("Density") +
  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),
    strip.text = element_text(size = axisTitle),
    strip.background = element_blank())

  # Each population figure
  each_fig <- mod %>%
  filter(species != "Perennial") %>%
  ggplot(aes(x = time, y = N, group = species, color = life_history, linetype = status)) +

```

```

geom_line(size = 1.5) +
facet_wrap(~model) +
  xlab("Time") +
  ylab("Density") +
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),
      strip.text = element_text(size = axisTitle),
      strip.background = element_blank())

# Output figure
print(plot_grid(tot_fig, each_fig, nrow = 2))

# Save output figure
pdf(filename)
print(plot_grid(tot_fig, each_fig, nrow = 2))
dev.off()

# Output dataframe
return(mod)
}

```

Original models (parameterization 1)

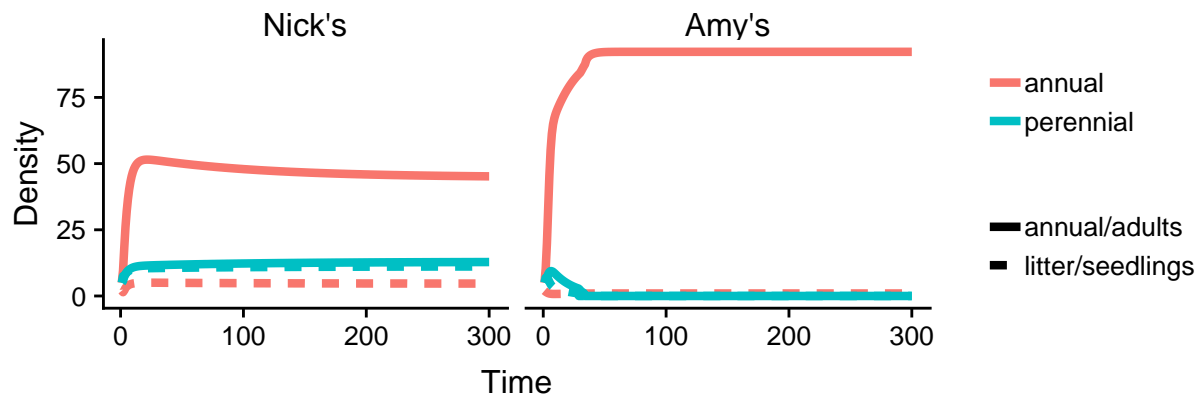
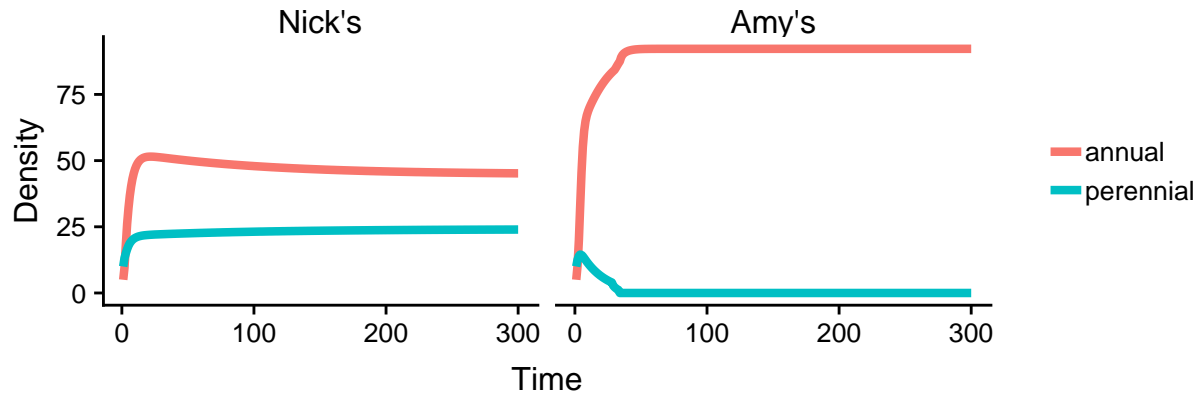
```

nmod <- simFun_n(params = params,
  NO.a = NO.a,
  NO.s = NO.s,
  NO.p = NO.p,
  LO = LO,
  simtime = years)

amod <- simFun_a(params = params,
  NO.a = NO.a,
  NO.s = NO.s,
  NO.p = NO.p,
  LO = LO,
  simtime = years,
  alt.comp = F,
  alt.per = F,
  alt.seed = F)

mod1 <- figfun(amod, nmod, "./microstegium_model_kortessis_comparison_output_orig.pdf")

```

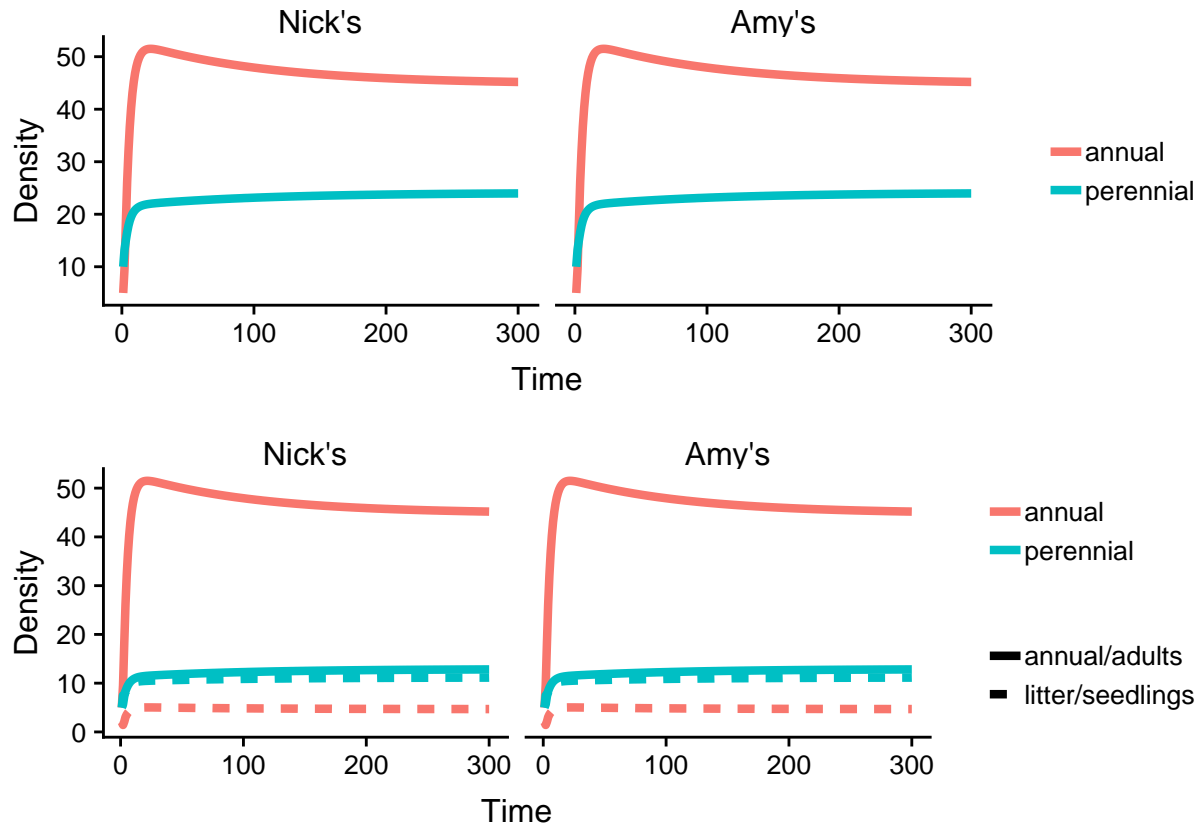



Make models equivalent (parameterization 2)

```
params2 <- params %>%
  mutate(value = case_when(symbol == "c.a" ~ 0.5,
                             symbol == "b" ~ 0.6,
                             TRUE ~ value))

amod2 <- simFun_a(params = params2,
  N0.a = N0.a,
  N0.s = N0.s,
  N0.p = N0.p,
  L0 = L0,
  simtime = years,
  alt.comp = T,
  alt.per = T,
  alt.seed = T)

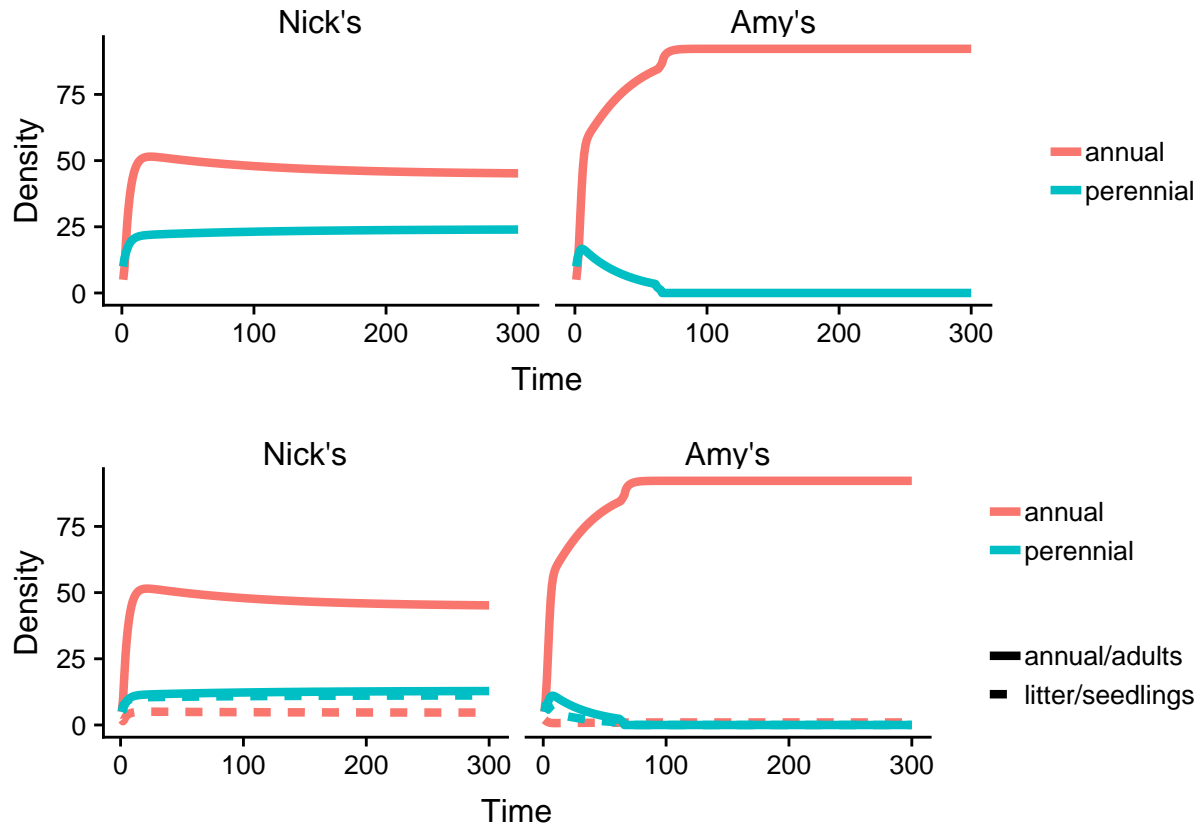
mod2 <- figfun(amod2, nmod, "./microstegium_model_kortessis_comparison_output_equal.pdf")
```



Use Amy's litter parameters (parameterization 3)

```
amod3 <- simFun_a(params = params,
  NO.a = NO.a,
  NO.s = NO.s,
  NO.p = NO.p,
  L0 = L0,
  simtime = years,
  alt.comp = T,
  alt.per = T,
  alt.seed = T)

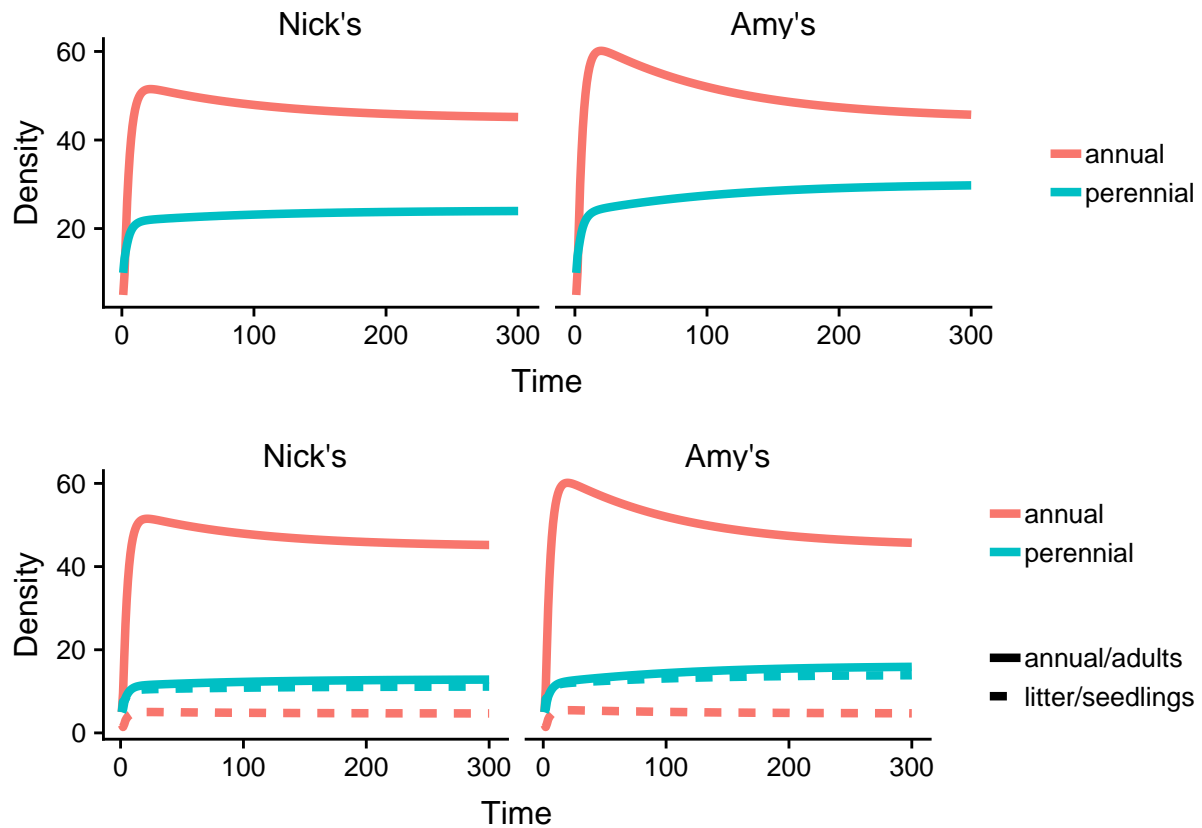
mod3 <- figfun(amod3, nmod, "./microstegium_model_kortessis_comparison_output_aklitter.pdf")
```



Use Amy's competition parameterization (parameterization 4)

```
amod4 <- simFun_a(params = params2,
  NO.a = NO.a,
  NO.s = NO.s,
  NO.p = NO.p,
  L0 = L0,
  simtime = years,
  alt.comp = F,
  alt.per = T,
  alt.seed = T)

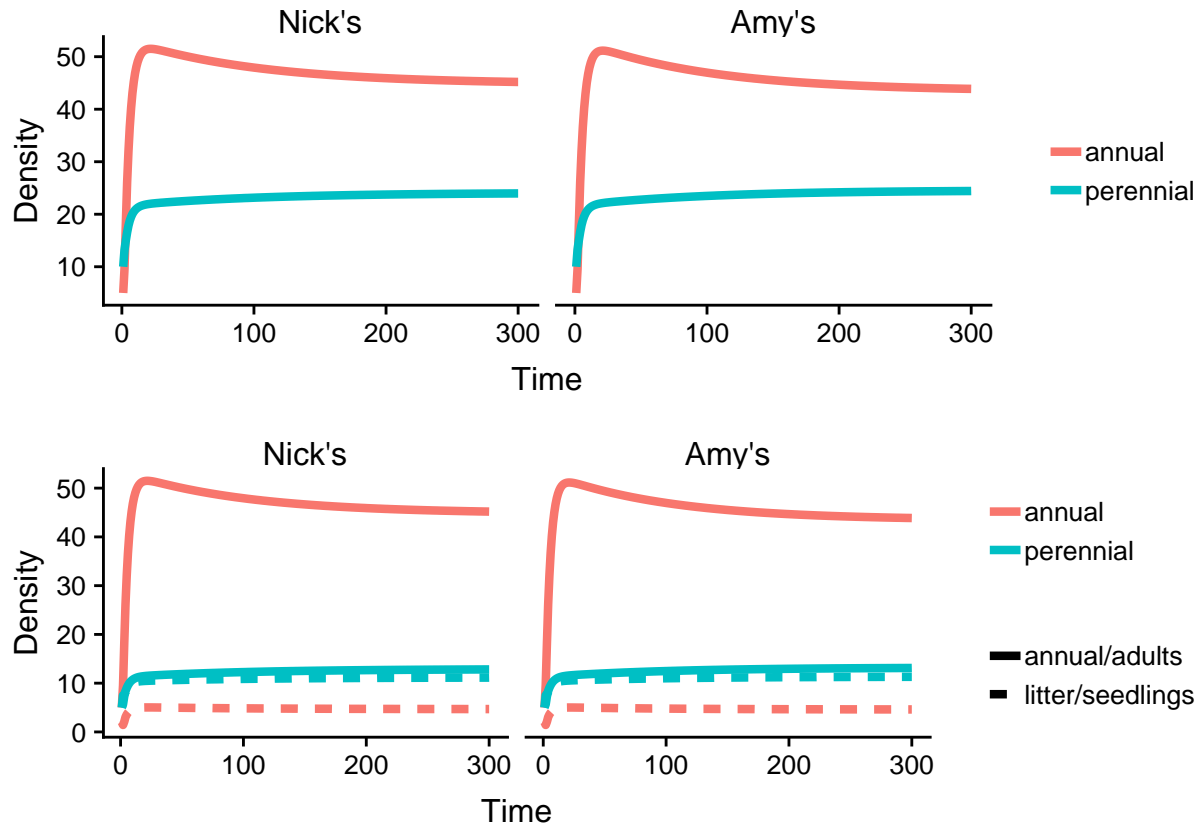
mod4 <- figfun(amod4, nmod, "./microstegium_model_kortessis_comparison_output_akcomp.pdf")
```



Use Amy's assumptions about first-year perennial seed production (parameterization 5)

```
amod5 <- simFun_a(params = params2,
  NO.a = NO.a,
  NO.s = NO.s,
  NO.p = NO.p,
  L0 = L0,
  simtime = years,
  alt.comp = T,
  alt.per = F,
  alt.seed = T)

mod5 <- figfun(amod5, nmod, "./microstegium_model_kortessis_comparison_output_akper.pdf")
```



Use Amy's assumption perennial survival before seed production (parameterization 6)

```
amod6 <- simFun_a(params = params2,
  NO.a = NO.a,
  NO.s = NO.s,
  NO.p = NO.p,
  L0 = L0,
  simtime = years,
  alt.comp = T,
  alt.per = T,
  alt.seed = F)

mod6 <- figfun(amod6, nmod, "./microstegium_model_kortessis_comparison_output_akseed.pdf")
```

