GUIDANCE DOCUMENT

Web Tools for Riparian and Aquatic Population Modeling

SERDP Project RC-2511

JANUARY 2020

Dr. David A. Lytle
Oregon State University

Dr. Jonathan Tonkin
University of Canterbury

Dr. Julian D. Olden
University of Washington

Dr. David M. Merritt
US Forest Service, Watershed, Fish, Wildlife, Air and Rare Plants

Dr. Laura McMullen
ICF International

Jane Rogosch
University of Washington

Distribution Statement A
This document has been cleared for public release
This report was prepared under contract to the Department of Defense Strategic Environmental Research and Development Program (SERDP). The publication of this report does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official policy or position of the Department of Defense. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the Department of Defense.
Web Tools for Riparian and Aquatic Population Modeling

This document assembles computer code, supporting data, and supplementary materials necessary to implement the flow-population models developed as part of RC-2511, Flow-population models for tracking non-stationary changes in riparian and aquatic ecosystems. Materials are presented separately for the three modeling structures pertaining to riparian vegetation, fish, and aquatic invertebrates. All materials are available free to the public online at the URLs below. The first section of this document presents and abbreviated summary of the materials with only brief descriptions and direct web links. The second section adds detailed descriptions of linked content.

DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

SUBJECT TERMS
Web Tools, Riparian and Aquatic Population Modeling, Flow-population Models, Non-stationary Changes, Riparian and Aquatic Ecosystems
WEB TOOLS FOR RIPARIAN AND AQUATIC POPULATION MODELING

RC-2511

Date: January 25, 2020

Authors
Dr. David A. Lytle, Department of Integrative Biology, Oregon State University
Dr. Jonathan Tonkin, University of Canterbury
Dr. Julian D. Olden, School of Aquatic and Fishery Sciences, University of Washington
Dr. David M. Merritt, US Forest Service, Watershed, Fish, Wildlife, Air and Rare Plants
Dr. Laura McMullen, ICF International
Jane Rogosch, PhD student, University of Washington
Abstract: This document assembles computer code, supporting data, and supplementary materials necessary to implement the flow-population models developed as part of RC-2511, *Flow-population models for tracking non-stationary changes in riparian and aquatic ecosystems*. Materials are presented separately for the three modeling structures pertaining to riparian vegetation, fish, and aquatic invertebrates. All materials are available free to the public online at the URLs below. The first section of this document presents and abbreviated summary of the materials with only brief descriptions and direct web links. The second section adds detailed descriptions of linked content.

Section 1: Summary of Web Tools and Data URLs

**Riparian vegetation model**

Main publications describing the methodology:
[https://doi.org/10.1002/eap.1528](https://doi.org/10.1002/eap.1528)

[https://www.nature.com/articles/s41559-017-0379-0](https://www.nature.com/articles/s41559-017-0379-0)

ShinyApp demo:
[https://jdtonkin.shinyapps.io/CotTam/](https://jdtonkin.shinyapps.io/CotTam/)
This is a simplified version of the riparian model that allows the user to directly adjust the drought frequency and project the model forward in time.
Vital rate estimation for xeroriparian shrub and hydoriparian tree:
An illustrative example of the process of obtaining species vital rates from the literature.

R code for implementing the riparian model:
https://figshare.com/articles/5-guild_riparian_flow-population_model/4652608

Sample hydrograph input data for riparian model (Maybell.csv):
https://figshare.com/articles/5-guild_riparian_flow-population_model/4652608

Fish population model

Primary publication describing the fish model methodology:
https://doi.org/10.1002/ecs2.2681

Fish model vital rate estimation:

R code for fish model:
https://doi.org/10.5281/zenodo.1309024

Test dataset for fish model:
https://zenodo.org/record/1309024#.XitmFhPYrVo
Invertebrate population model

Primary publication describing the methodology:
https://doi.org/10.1111/ele.12866

Mathematical proof of the time-varying logistic model:

Methodology for obtaining vital rates for the invertebrate model:

R code for implementing the invertebrate model:
https://doi.org/10.6084/m9.figshare.5476993
Section 2: Detailed resource descriptions

Riparian vegetation model

Main publications describing the methodology:
https://doi.org/10.1002/eap.1528

https://www.nature.com/articles/s41559-017-0379-0

ShinyApp demo:
https://jdtonkin.shinyapps.io/CotTam/
This is a simplified version of the riparian model that allows the user to directly adjust the drought frequency and project the model forward in time.

Vital rate estimation for xeroriparian shrub and hydoriparian tree:
An illustrative example of the process of obtaining species vital rates from the literature.

Xeroriparian shrub and hydoriparian tree possess similar vital rates in terms of response to flooding, timing of seedset, and other vital rates. We review the literature pertaining to these two guilds and present the relevant data here. For xeroriparian shrub we used chronosequence studies along the upper and lower Colorado River, where stem thinning rate averaged 2,317 (±394) stems ha⁻¹ yr⁻¹ along the upper Colorado
River and 3,257 (±937) stems ha\(^{-1}\) yr\(^{-1}\) along the lower Colorado River (Merritt and Shafroth 2012). Self-thinning relations were: \(S = 130,790-2,317\) (age) and \(S = 167,781-3,257\) (age). Stands older than 50 years had average stem densities of 12,300 and 16,800 stems ha\(^{-1}\) along the upper and lower Colorado River, respectively. There was no significant difference in the thinning rate between the upper and lower Colorado River (\(P = 0.1\)), and the overall thinning rate of xeroriparian shrub was 2,455 ±312 stems ha\(^{-1}\) yr\(^{-1}\) over the 70 years modeled. The relationship between stand age and xeroriparian shrub stem density (ST) along both upper and lower Colorado River sites was \(ST = 135,478-2,455\) (age) (\(r^2 = 0.27; P < 0.0001\)). Using data from Horton and Clark (2001) and Shafroth et al. (1998) we estimated the relationship between stage decline and seedling recruitment for xeroriparian shrub as \(g(h) = 0.92\exp[-0.5\times((h-1.8)/3.4)^2]\) for \(h \geq 0\); 0 otherwise. The shape of this function is similar to that of hydoriparian tree (Lytle and Merritt 2004), but reflects the fact that xeroriparian shrub seedlings are more tolerant of static water levels and rapid water declines than hydoriparian tree seedlings.

Hydoriparian tree and xeroriparian shrub have several similarities in terms of life history attributes and regeneration niche, as well as some important differences. Species in both guilds (e.g., *Populus deltoides* and *Tamarix ramosissima*, respectively) require bare, moist freshly exposed substrate for short-lived aerial or water dispersed seeds to germinate and become established (Merkel and Hopkins 1957, Warren and Turner 1975, Fenner et al. 1984). Flood-created bare patches and areas of fresh sediment deposition are common sites for recruitment; neither guild typically recruit into heavily vegetated or shaded areas (Braatne et al. 1996, Scott et al. 1996, Cooper et al. 1999). Typically, plant establishment occurs as part of the processes of channel narrowing, point bar development due to channel meandering, or overbank deposition of sediment (Scott et al. 1996, Cooper et al. 2003). If viable seeds reach such sites, hydrologic conditions are conducive to seedling survival, and subsequent disturbance does not remove or bury individuals, establishment may occur.

Timing of flowering and seed dispersal phenology differ significantly for the hydoriparian tree and xeroriparian shrub guilds in the climate of the Colorado Plateau and in much of the western U.S. (Warren and Turner 1975, Cooper et al. 1999). In
warm climates, xeroriparian shrub may flower as early as its second year of growth and flowering may occur multiple times during the growing season (Merkel and Hopkins 1957, Horton et al. 1960). Older, larger individuals may produce several hundred thousand seeds during a single season (Merkel and Hopkins 1957). Hydroriparian tree reaches reproductive stage later (5-10 years) and flowers only once per growing season, yet females may produce tens of thousands of seeds per growing season (Karrenberg and Suter 2003). Furthermore, xeroriparian shrub has bisexual flowers, so each individual produces seeds in contrast to dioecious hydroriparian tree, which has a lower ratio of seed producing individuals to individual plants (Warren and Turner 1975).

Our representative species for the hydroriparian guild, *Populus fremontii*, was found to release seeds over a six week period beginning in late-June along the Yampa River (Cooper et al. 1999). By contrast, our representative for xeroriparian shrub, *Tamarix*, began releasing seed in mid-July, seed rain peaked in mid-August, and seeds were still being dispersed in mid-September during the years measured (Cooper et al. 1999). Warren and Turner (1975) found that *Populus* dispersal occurred earlier than *Tamarix*, was of a shorter duration, and had almost ceased by the time the flowering season for *Tamarix* began along two rivers in Arizona, U.S.A.

*Tamarix* seed density in areas of establishment has been shown to be as high as \(~5000\) seedlings m\(^{-2}\) (Cooper et al. 1999) to \(170,000\) seedlings m\(^{-2}\) in dense *Tamarix* stands (Warren and Turner 1975). In germination trials, *Tamarix* seed viability ranged from 76\% at the time of harvest to 40\% after four months of cold storage (Merkel and Hopkins 1957), but seed viability in the field lasts only a few weeks (Horton et al. 1960). Laboratory trials indicate that viability ranges from 75\% (Moss 1938) to greater than 90\% (Van Splunder et al. 1995, Karrenberg and Suter 2003) for various species of *Populus*.

The combination of limited seed viability and differences in the separation in seed release timing of seed release may result in spatial separation in hydroriparian tree and xeroriparian invasive shrub recruitment sites, though mixed stands are not uncommon may occur due to dispersal overlap during the middle of some growing seasons. Hypocotyl extension and primary root growth occur in moist soils and at this stage
seedlings are very vulnerable to desiccation. *Populus* seedling root growth rates have been shown to average 0.6-1.3 cm/day, resulting in 72-162 cm of root growth by the end of their first season (Fenner et al. 1984, Mahoney and Rood 1998, Horton and Clark 2001), however, it has been reported that *Populus* seedlings can survive ground water decline rates of 2-4 cm day⁻¹ (Mahoney and Rood 1991, Segelquist et al. 1993). These values may more than double in finer soils with higher water-holding retention capacity soils (Cooper et al. 1999). *Populus* root growth and leaf area were found to be highest with steady shallow water tables and declined as a function of water table drawdown rate in rhizopod experiments (Mahoney and Rood 1991). *Tamarix* root growth rates were 1.1 cm d⁻¹ in field studies in Arizona, U.S.A. (Merkel and Hopkins 1957). *Tamarix* survival was 86-92% across treatments involving lowering water tables 0, 1, 2, and 4 cm/d, though biomass declined with increasing rates of decline (Horton and Clark 2001). *Tamarix* root growth rate was highest in 1 cm d⁻¹ water table drawdowns and *Tamarix* root length averaged 160 cm after 42 days of growth (about double that for *Salix gooddingii*) (Horton and Clark 2001). *Tamarix* seedlings are more tolerant of groundwater declines than are *S. gooddingii* and *Populus*, because it is known to utilize water from both phreatic sources and unsaturated soils (Everitt 1980, Busch and Smith 1995). Higher drought tolerance may enable *Tamarix* seedlings to persist in dry soils where *Salix* and *Populus* seedlings are unable (Cooper et al. 1999, Horton and Clark 2001).

Seedlings in both guilds are vulnerable to flow-related scour and deposition as well as wind abrasion and dune-burial during the first several months of growth. Due to *Tamarix* seedlings’ prostrate growth form in the seedling stage, it may render it more susceptible than *Populus* to burial (Levine and Stromberg 2001), but may be more resistant to scour due to the high tensile strength of the roots (De Baets et al. 2008). *Tamarix* is more vulnerable to inundation and anoxia in seedling and sapling stages than *Populus* (Bhattacharjee et al. 2006). Sher et al. (2000) and Sher and Marshall (2003) found that *Tamarix* was competitively suppressed by *Populus* mixed stand experiments, suggest that flooding puts native *Populus* at a competitive advantage over *Tamarix*. Degree of shade had no effect on survival of *Populus* seedlings under well-
watered conditions, but dramatically effected survival under drought conditions (43% survival in full sun; 0.03% survival in 4% sun) (Cooper et al. 1999). Interspecific competition contributed significantly to mortality over a range of light levels in field experimentation (Cooper et al.1999). Because *Tamarix* maximum aboveground height is less than *Populus*, competition for light and the negative effects of shading remain inhibitive for older age-classes of *Tamarix* (Dewine and Cooper 2008).

References


R code for implementing the riparian model:
https://figshare.com/articles/5-guild_riparian_flow-population_model/4652608
For code to run, it requires the accompanied 'Maybell.csv' file in the same directory.

# 5-guild riparian plant flow-population model
# Date: Feb 2017
# Authors: Jonathan Tonkin, David Lytle
# Emails: jdtonkin@gmail.com, lytleda@oregonstate.edu

# Running the code as is produces a 250 year projection of the five guilds at the current natural flow regime. To alter the flow regime, change 'outerreps' to however severe you want flow alteration to be and change the settings inside the 'FLOW ALTERATIONS' section. Changing 'outerreps' to 84 alters flood or droughts from natural flow to 100% modified.

# Note: In the paper, we refer to five guilds: HT (Hydroriparian Tree), XS (Xeroriparian Shrub), HS (Hydroriparian Shrub), MM (Mesoriparian Meadow), and DS (Desert Shrub).
# Here, we use different names, with local examples as follows:
# HT: Cottonwood. Includes anything with 'C' or 'Cot'. e.g. 'Cgraph', 'Crep', 'DomC'
# XS: Tamarisk. Includes anything with 'T' or 'Tam'. e.g. 'Tgraph', 'Trep', 'DomT'
# HS: Willow. Includes anything with 'W'. e.g. 'Wgraph', 'Wrep'
# MM: Meadow. Includes anything with 'M'. e.g. 'Mgraph', 'Mrep'
# DS: Sagebrush. Includes anything with 'S'. e.g. 'Sgraph', 'Srep'

# Required libraries
library(ggplot2)
library(tidyr)
library(dplyr)

# SETUP

rm(list = ls()) # clearing the workspace
count <- 250 # number of years to project simulations (inner loop)
burnin = 100  # number of years to discard as burn in during long term mean estimation
outerreps <- 1  # number of iterations for outer loop that alters drought/flood frequency
replicates <- 100  # number of replicate projections to run (mid loop)

# FLOW

# Maybell flow data 1916-1998, 83 years continuous
flowdata <- read.csv("maybell.csv")
str(flowdata)
head(flowdata)
# flooddates is peak dates of all floods (Oct 1 = 1)
# basedates is baseflow dates

# FLOOD REGIME DEFINITIONS

# CALCULATION OF p<j> and p<tam>  
# Prop. of possible recruitment days that occur between Q and return to baseflow at 700cfs

cseedfirst = 260  # COT: first date of seedset, June 17th or 260
cseedlast = 298  # COT: last date of seedset, 298 or July 25.
tseedfirst = 280  # TAM: first date of seedset, which is July 7th or 280
tseedlast = 336  # TAM: last date of seedset, 336 or 1 Sept.

# FUNCTIONS FOR INCLUDING INSIDE LOOPS

# Assign either the first date of cottonwood seedset or the first date of flooding
# Cottonwood
csf_func <- function(x) {
  ifelse(cseedfirst > x, cseedfirst, x)
}
cfirstdate <- csf_func(flowdata$flooddates)

# Assign either the last date of cottonwood seedset or the date that baseflow occurs
csl_func <- function(x) {
  ifelse(cseedlast > x, x, cseedlast)
}
clastdate <- csl_func(flowdata$basedates)

# Calculate the number of days in the seedset period
cdifference <- clastdate - cfirstdate

# Check to make sure value is positive, assigns 0 otherwise
gzero_func <- function(x) {
  ifelse(x > 0, x, 0)
}

# Length of season for seedset - with negative values removed
cseasonlength <- gzero_func(cdifference)

# Proportion of season available
cproportion <- cseasonlength/(cseedlast - cseedfirst)

# Now repeat all this to get p<tam>
tsf_func <- function(x) {
  ifelse(tseedfirst > x, tseedfirst, x)
}
tfirstdate <- tsf_func(flowdata$flooddates)
tsl_func <- function(x) {
  ifelse(tseedlast > x, x, tseedlast)
}
tlastdate <- tsl_func(flowdata$basedates)
tdifference <- tlastdate - tfirstdate
tseasonlength <- gzero_func(tdifference)
tproportion <- tseasonlength/(tseedlast - tseedfirst)

# FLOOD THRESHOLD FUNCTIONS

# Currently these are the same for cot and tam
# flowdata$floodmag - vector containing peak flood magnitude

# Magnitude of peak flow over which is considered a mortality causing flood event
mortcutoff = 9888 # This is in CFS, as are the Maybell data points
# Floods above 280 cms 9888 cfs cause mortality, below cause none
# Same for drought (threshold below rather than above)
droughtcutoff = 7416 # floods above 210 cms / 7416cfs do not cause
drought mortality

# Convert peak discharge values into a vector of floods/no floods,
# 1 = an above threshold flood (some scouring), 0 = below threshold
# (no scouring)
bigflood_func <- function(x) {
  ifelse(x > mortcutoff, 1, 0)
}

# Another similar vector for droughts
# Both of these appear in the functions for vital rates and fecundity,
# below
drought_func <- function(x) {
  ifelse(x < droughtcutoff, 1, 0)
}

# Vector containing 1's for years that do not have a flood, e.g.
# normal and drought years
nonflood_func <- function(x) {
  ifelse(x > mortcutoff, 0, 1)
}

# ITERATION PARAMETERS
# Setting up arrays/vectors to fill with data from loops

# Inner loop details

# 'count' - number of years to project simulations (inner loop)
# Output of no. ind. for each age class for each year projected into
# the future
# An array with 6 columns (each age class) and however many rows there
# are years projected
Coutput <- array(0, dim = c(count, 6)) # Cottonwood
Toutput <- array(0, dim = c(count, 6)) # Tamarisk
Woutput <- array(0, dim = c(count, 6)) # Willow
Soutput <- array(0, dim = c(count, 6)) # Sagebrush
Moutput <- array(0, dim = c(count, 6)) # Meadow
# Total cottonwood pop. size as % of K
# This is the total space occupied by this species in cottonwood seedling units
Cspaceoutput <- numeric(length = count) # Cottonwood
Tspaceoutput <- numeric(length = count) # Tamarisk
Wspaceoutput <- numeric(length = count) # Willow
Sspaceoutput <- numeric(length = count) # Sagebrush
Mspaceoutput <- numeric(length = count) # Meadow

# Flood and drought settings for each year projected into the future (i.e. 0 or 1)
floodoutput <- numeric(length = count) # flood
droughtoutput <- numeric(length = count) # drought
nonfloodoutput <- numeric(length = count) # nonflood
normaloutput <- numeric(length = count) # normal

# Total pop. size as % of K WITHOUT SEEDLINGS for each year projected into future
# contains %K of each guild except for seedlings for each year of projection
Cnonseedling <- numeric(length = count)
Tnonseedling <- numeric(length = count)
Wnonseedling <- numeric(length = count)
Snonseedling <- numeric(length = count)
Mnonseedling <- numeric(length = count)

# No. ind. at stages as per Merritt and Poff 2010
# Vector of cot age class 5 for each year projected
DomC <- numeric(length = count) # these will record the NUMBER of individuals in stg 5,
# which are 5-10 year olds
DomT <- numeric(length = count) # same, for stages 4, although this is 7-15 year olds

# Mid loop details -----------------------------------------------
# 'replicates' - number of replicate projections to run (mid loop)

# Mean values for each rep. run over the period specified from burnin to end of projection
# Mean density of each guild WITHOUT seedlings included - each replicate run
Crep <- numeric(length = replicates)
Trep <- numeric(length = replicates)
Wrep <- numeric(length = replicates)
Srep <- numeric(length = replicates)
Mrep <- numeric(length = replicates)

# Mean density of each guild WITH seedlings included
Crep_all <- numeric(length = replicates)
Trep_all <- numeric(length = replicates)
Wrep_all <- numeric(length = replicates)
Srep_all <- numeric(length = replicates)
Mrep_all <- numeric(length = replicates)

# Mean of DomC for each of the replicate runs
DomCrep <- numeric(length = replicates) # to record output of DomC from flow scenarios
DomTrep <- numeric(length = replicates)

# Outer loop details -----------------------------------------------
# 'outerreps' - number of iterations for outer loop that alters
drought/flood frequency
# Results of flow mod scenarios. Not useful unless simulating changes
to flow regime
# This is the mean of each flow mod. setting for the full burnin->end
# of projection period

# No seedlings for each of the flow mod settings
Cgraph <- numeric(length = outerreps)
Tgraph <- numeric(length = outerreps)
Mgraph <- numeric(length = outerreps)
Wgraph <- numeric(length = outerreps)
Sgraph <- numeric(length = outerreps)

# All incl. seedlings for each of the flow mod settings
Cgraph_all <- numeric(length = outerreps)
Tgraph_all <- numeric(length = outerreps)
Mgraph_all <- numeric(length = outerreps)
Wgraph_all <- numeric(length = outerreps)
Sgraph_all <- numeric(length = outerreps)

# Mean of DomC for each of the flow mod settings
DomCrep_graph <- numeric(length = outerreps)
DomTrep_graph <- numeric(length = outerreps)

# Proportion of flow years in each flow mod setting
droughtgraph <- numeric(length = outerreps) # droughts
floodgraph <- numeric(length = outerreps) # floods
nonfloodgraph <- numeric(length = outerreps) # nonfloods
normalgraph <- numeric(length = outerreps) # normal years

# Setting flow scenario change to none to begin with
# The outer loop iterates these one step at a time until it reaches 'outerreps'
droughtchanged = floodchanged = 0

# RECRUITMENT AS FUNCTION OF FLOOD DECLINE RATE ------------------------------

# COT: vector containing the NEGATIVE VALUE OF slopes of all declining limbs of floods
flooddecline <- -flowdata$slopecot
# COT: drawdown survival, a lognormal FUNCTION.
# Positive values of h indicate receding water
cdds_func <- function(x) {
  .944 * exp(-.5 * ((log(x/1.279))/.987)^2)
}
decline <- cdds_func(flooddecline) # flood decline is the h value from paper

# TAM: vector containing NEGATIVE VALUE OF slopes of all declining limbs of floods
tflooddecline <- -flowdata$slopetam
# TAM: drawdown survival, a lognormal FUNCTION
tdds_func <- function(x) {
  0.917 * exp(-.5 * ((x - 1.8)/3.4)^2)
}
tdecline <- tdds_func(tflooddecline)

# Checks to see if at least one adult is present
adult_func <- function(x) {
  ifelse(x > .99999, 1, 0)
}
# Keeps FC6 from dividing by zero by substituting an arbitrary nonzero number that will get multiplied by zero later anyway during matrix multiplication
nonind <- function(x) {
  ifelse(x == 0, 666, x)
}

# checkpos makes sure that the K-occupied term is positive, assigns 0 if not
checkpos <- function(x) {
  ifelse(x < 0, 0, x)
}

# Quasi extinction threshold of 1, keeps pop from asymptoting infinitely to zero
quasi <- function(x) {
  ifelse(x < 1, 0, x)
}

# Quasi rescue function that keeps species from disappearing, e.g. sagebrush that can encroach from uplands
quasiten <- function(x) {
  ifelse(x < 10, 10, x)
}

# OUTER LOOP
for(zim in 1:outerreps) {
  bigflood <- bigflood_func(flowdata$floodmag)
  drought <- drought_func(flowdata$floodmag)
  nonflood <- nonflood_func(flowdata$floodmag)

# IMPORTANT: For modifying bigflood/drought/normal - you need to be
careful to modify them
# simultaneously, otherwise you can get simultaneous drought and flood
# years
# e.g. #1 incr. floods: add 1s to bigflood, but also add 0s to both
drought and nonflood.
# e.g. #2 homogenizing flows: add 0s to bigflood and drought and add
1s to nonflood.
# e.g. #3 increasing drought: add 1s to drought and nonflood and add
0s to bigflood.

# Comment/uncomment the following to set up these scenarios

# ifelse(floodchanged == 0, bigflood, bigflood[1:floodchanged] <- 0)
# '0' eliminates floods in Maybell vector going from year 1 to
# floodchanged
# Floodchanged is how many flood years to remove
# To CREATE floods use = 1

# ifelse(droughtchanged == 0, drought, drought[1:droughtchanged] <- 0)
# '0' eliminates droughts in Maybell vector going from year 1 to
droughtchanged
# To CREATE droughts use = 1

# ifelse(floodchanged == 0, nonflood, nonflood[1:floodchanged] <- 1)

# ----------------------------------------------------------------------
# FLOW ALTERATIONS end
# ----------------------------------------------------------------------

# MIDDLE LOOP
# Middle loop uses iterator "rep" to get "replicates" number of runs
# for averaging
for(rep in 1:replicates) {

```
# VITAL RATES

# K is total area available to cottonwood or tamarisk initially calculated as total area occupied by cottonwood.

# VITAL RATES - cottonwood

# Stage specific densities

denC1 <- 350
denC2 <- 10
denC3 <- 1
denC4 <- .91
denC5 <- .6
denC6 <- .12

# "Self thinning" rates, or equivalency rules, for stage transitions

bC1 <- denC2/denC1
bC2 <- denC3/denC2
bC3 <- denC4/denC3
bC4 <- denC5/denC4
bC5 <- denC6/denC5

# Baseline maturation probability, aC6 (adult senescence rate)

aC1 <- 1
aC2 <- 1
aC3 <- 1
aC4 <- 1
aC5 <- .167
aC6 <- .03

# Flood mortality in a flood year

SC1 <- .97
SC2 <- .33
SC3 <- .224
SC4 <- .19
SC5 <- .073
SC6 <- .02

# Drought mortality in a drought year
DC1 <- .49
DC2 <- .16
DC3 <- .083
DC4 <- .05
DC5 <- .05
DC6 <- .05

# Initial area in m². Here, it is calculated based ONLY on cottonwood.
areaC1 <- 11816
areaC2 <- 11816
areaC3 <- 11144
areaC4 <- 11144
areaC5 <- 13819
areaC6 <- 59605

K <- (areaC1 + areaC2 + areaC3 + areaC4 + areaC5 + areaC6) * denC1

# VITAL RATES - TAMARISK
---------------------------

# K is common to both cot and tam

# Stage specific densities, number per m²
denT1 <- 400
denT2 <- 29
denT3 <- 4.5
denT4 <- 1.4
denT5 <- 1.3
denT6 <- 1.3

# "Self thinning" rates, or equivalency rules, for stage transitions
bT1 <- denT2/denT1
bT2 <- denT3/denT2
bT3 <- denT4/denT3
bT4 <- denT5/denT4
bT5 <- denT6/denT5

# Baseline maturation probability, aT6 (adult senescence rate)
AT1 <- 1
AT2 <- 1
AT3 <- .25
AT4 <- .11
aT5 <- .07
aT6 <- .05

# Flood mortality in a flood year
ST1 <- .9
ST2 <- .55
ST3 <- .25
ST4 <- .05
ST5 <- .01
ST6 <- .01

# Drought mortality in a drought year
DT1 <- .5
DT2 <- .15
DT3 <- .05
DT4 <- .025
DT5 <- .025
DT6 <- .025

# Initial area in m2. Not included in initial K calculation
areaT1 <- 1000
areaT2 <- 0
areaT3 <- 0
areaT4 <- 0
areaT5 <- 0
areaT6 <- 0

# VITAL RATES - willow

# Stage specific densities, number per m2
# THESE NUMBERS ARE ARBITRARY, but reflect a final adult size/spacing
# of 1 m2 per plant
denW1 <- 350
denW2 <- 35
denW3 <- 1
denW4 <- 1
denW5 <- 1
denW6 <- 1

# "Self thinning" rates, or equivalency rules, for stage transitions
bW1 <- denW2/denW1
bW2 <- denW3/denW2
bW3 <- denW4/denW3
bW4 <- denW5/denW4
bW5 <- denW6/denW5

# Baseline maturation probability, aW6 is adult senescence rate
# Only aW6 is different from cottonwood (faster)
aW1 <- 1
aW2 <- 1
aW3 <- 1
aW4 <- 1
aW5 <- .167
aW6 <- .01

# Flood mortality in a flood year. HALF THAT OF COTTONWOOD
SW1 <- .49
SW2 <- .17
SW3 <- .11
SW4 <- .10
SW5 <- .04
SW6 <- .01

# Drought mortality in a drought year
# Basically, SURVIVORSHIP rate is half that of cottonwood: convert mortality to
# survivorship, divide by two, convert back to mortality
DW1 <- .75
DW2 <- .58
DW3 <- .54
DW4 <- .53
DW5 <- .51
DW6 <- .51

# Initial area in m2. Not included in initial K calculation
areaW1 <- 100
areaW2 <- 100
areaW3 <- 100
areaW4 <- 100
areaW5 <- 100
areaW6 <- 100
# VITAL RATES - sagebrush ---------------------------------

# Stage specific densities, number per m2
# THESE NUMBERS ARE ARBITRARY, but reflect an adult size/spacing of 1 m2
denS1 <- 350
denS2 <- 35
denS3 <- 1
denS4 <- 1
denS5 <- 1
denS6 <- 1

# "Self thinning" rates, or equivalency rules, for stage transitions
bS1 <- denS2/denS1
bS2 <- denS3/denS2
bS3 <- denS4/denS3
bS4 <- denS5/denS4
bS5 <- denS6/denS5

# Baseline maturation probability, aS6 is adult senescence rate.
# SAME AS COTTONWOOD
aS1 <- 1
aS2 <- 1
aS3 <- 1
aS4 <- 1
aS5 <- .167
aS6 <- .03

# Flood mortality in a flood year
# Taken by taking survivorship of cot (1-S), dividing by 2, and converting back to mort.
# So, flood SURVIVORSHIP is 1/2 that of cottonwood
SS1 <- .99
SS2 <- .67
SS3 <- .61
SS4 <- .60
SS5 <- .54
SS6 <- .51

# Drought MORTALITY in a drought year. ONE HALF COTTONWOOD RATES
DS1 <- .24
DS2 <- .08
DS3 <- .042
DS4 <- .025
DS5 <- .005
DS6 <- .005

# Initial area in m2. Not included in initial K calculation
areaS1 <- 100
areaS2 <- 100
areaS3 <- 100
areaS4 <- 100
areaS5 <- 100
areaS6 <- 100

# VITAL RATES - Meadow -----------------------------------------------
# Stage specific densities, number per m2
# THESE NUMBERS ARE ARBITRARY, but reflect an adult size/spacing of 1 m2
denM1 <- 350
denM2 <- 35
denM3 <- 1
denM4 <- 1
denM5 <- 1
denM6 <- 1

# "Self thinning" rates, or equivalency rules, for stage transitions
bM1 <- denM2/denM1
bM2 <- denM3/denM2
bM3 <- denM4/denM3
bM4 <- denM5/denM4
bM5 <- denM6/denM5

# Baseline maturation probability, aM6 is adult senescence rate. SAME AS COTTONWOOD
aM1 <- 1
aM2 <- 1
aM3 <- 1
aM4 <- 1
aM5 <- .167
aM6 <- .03
# Flood mortality in a flood year
# Taken by taking survivorship of cot (1-S), dividing by 2, and converting back to mort.
# FOR STAGES 1-3 ONLY
# So, flood SURVIVORSHIP is 1/2 that of cottonwood for those stages, reflecting instability
# of meadow habitats in highly flood-prone situations.
SM1 <- .99
SM2 <- .67
SM3 <- .61
SM4 <- .19
SM5 <- .073
SM6 <- .02

# Drought MORTALITY in a drought year
# ONE HALF COTTONWOOD RATES in stages 4-6 only, reflecting greater drought tolerance in
# established meadows and less groundwater dependence
DM1 <- .49
DM2 <- .16
DM3 <- .083
DM4 <- .025
DM5 <- .005
DM6 <- .005

# Initial area in m2. Not included in initial K calculation
areaM1 <- 100
areaM2 <- 100
areaM3 <- 10
areaM4 <- 100
areaM5 <- 100
areaM6 <- 100

# Maybell area in m2, as established from initial cottonwood occupancy
# K is the total area available for cot OR tam expressed in cottonwood seedlings per m2
NC <- c(areaC1 * denC1,
        areaC2 * denC2,
        areaC3 * denC3,
        areaC4 * denC4,
        areaC5 * denC5,
areaC6 * denC6)
# NC gives the total number of individuals for each age class.
# Initially here, this is found by multiplying the number of m2
occupied by a given class
# by the the density per m2

NT <- c(areaT1 * denT1,
    areaT2 * denT2,
    areaT3 * denT3,
    areaT4 * denT4,
    areaT5 * denT5,
    areaT6 * denT6)

NW <- c(areaW1 * denW1,
    areaW2 * denW2,
    areaW3 * denW3,
    areaW4 * denW4,
    areaW5 * denW5,
    areaW6 * denW6)

NS <- c(areaS1 * denS1,
    areaS2 * denS2,
    areaS3 * denS3,
    areaS4 * denS4,
    areaS5 * denS5,
    areaS6 * denS6)

NM <- c(areaM1 * denM1,
    areaM2 * denM2,
    areaM3 * denM3,
    areaM4 * denM4,
    areaM5 * denM5,
    areaM6 * denM6)

# Inner loop

for(i in 1:count) {

    y = sample(nrow(flowdata), 1)
    # y is a random number within the length of the flow data to randomly
    select a year from
# the 'bigflood' and 'drought' vector
# in this case anything between 1 and 83 in the maybell data

# VITAL RATE DEFINITIONS: cottonwood ----------------------------------
# G is prob. of transition to next stage
# P is prob. of remaining in that stage
GC1 <- aC1 * bC1 * (1 - bigflood[y] * SC1) * (1 - drought[y] * DC1)
GC2 <- aC2 * bC2 * (1 - bigflood[y] * SC2) * (1 - drought[y] * DC2)
GC3 <- aC3 * bC3 * (1 - bigflood[y] * SC3) * (1 - drought[y] * DC3)
GC4 <- aC4 * bC4 * (1 - bigflood[y] * SC4) * (1 - drought[y] * DC4)
GC5 <- aC5 * bC5 * (1 - bigflood[y] * SC5) * (1 - drought[y] * DC5)
PC5 <- (1 - aC5) * (1 - bigflood[y] * SC5) * (1 - drought[y] * DC5)
PC6 <- (1 - aC6) * (1 - bigflood[y] * SC6) * (1 - drought[y] * DC6)

# VITAL RATE DEFINITIONS: tamarisk -----------------------------------
GT1 <- aT1 * bT1 * (1 - bigflood[y] * ST1) * (1 - drought[y] * DT1)
GT2 <- aT2 * bT2 * (1 - bigflood[y] * ST2) * (1 - drought[y] * DT2)
GT3 <- aT3 * bT3 * (1 - bigflood[y] * ST3) * (1 - drought[y] * DT3)
GT4 <- aT4 * bT4 * (1 - bigflood[y] * ST4) * (1 - drought[y] * DT4)
GT5 <- aT5 * bT5 * (1 - bigflood[y] * ST5) * (1 - drought[y] * DT5)
PT3 <- (1 - aT3) * (1 - bigflood[y] * ST3) * (1 - drought[y] * DT3)
PT4 <- (1 - aT4) * (1 - bigflood[y] * ST4) * (1 - drought[y] * DT4)
PT5 <- (1 - aT5) * (1 - bigflood[y] * ST5) * (1 - drought[y] * DT5)
PT6 <- (1 - aT6) * (1 - bigflood[y] * ST6) * (1 - drought[y] * DT6)

# VITAL RATE DEFINITIONS: willow -------------------------------------
GW1 <- aW1 * bW1 * (1 - bigflood[y] * SW1) * (1 - drought[y] * DW1)
GW2 <- aW2 * bW2 * (1 - bigflood[y] * SW2) * (1 - drought[y] * DW2)
GW3 <- aW3 * bW3 * (1 - bigflood[y] * SW3) * (1 - drought[y] * DW3)
GW4 <- aW4 * bW4 * (1 - bigflood[y] * SW4) * (1 - drought[y] * DW4)
GW5 <- aW5 * bW5 * (1 - bigflood[y] * SW5) * (1 - drought[y] * DW5)
PW5 <- (1 - aW5) * (1 - bigflood[y] * SW5) * (1 - drought[y] * DW5)
PW6 <- (1 - aW6) * (1 - bigflood[y] * SW6) * (1 - drought[y] * DW6)

# VITAL RATE DEFINITIONS: sagebrush ----------------------------------
GS1 <- aS1 * bS1 * (1 - bigflood[y] * SS1) * (1 - drought[y] * DS1)
GS2 <- aS2 * bS2 * (1 - bigflood[y] * SS2) * (1 - drought[y] * DS2)
GS3 <- aS3 * bS3 * (1 - bigflood[y] * SS3) * (1 - drought[y] * DS3)
GS4 \leftarrow aS4 \cdot bS4 \cdot (1 - \text{bigflood}[y] \cdot SS4) \cdot (1 - \text{drought}[y] \cdot DS4) \\
GS5 \leftarrow aS5 \cdot bS5 \cdot (1 - \text{bigflood}[y] \cdot SS5) \cdot (1 - \text{drought}[y] \cdot DS5) \\
PS5 \leftarrow (1 - aS5) \cdot (1 - \text{bigflood}[y] \cdot SS5) \cdot (1 - \text{drought}[y] \cdot DS5) \\
PS6 \leftarrow (1 - aS6) \cdot (1 - \text{bigflood}[y] \cdot SS6) \cdot (1 - \text{drought}[y] \cdot DS6)

# VITAL RATE DEFINITIONS: meadow -------------------------------------

GM1 \leftarrow aM1 \cdot bM1 \cdot (1 - \text{bigflood}[y] \cdot SM1) \cdot (1 - \text{drought}[y] \cdot DM1) \\
GM2 \leftarrow aM2 \cdot bM2 \cdot (1 - \text{bigflood}[y] \cdot SM2) \cdot (1 - \text{drought}[y] \cdot DM2) \\
GM3 \leftarrow aM3 \cdot bM3 \cdot (1 - \text{bigflood}[y] \cdot SM3) \cdot (1 - \text{drought}[y] \cdot DM3) \\
GM4 \leftarrow aM4 \cdot bM4 \cdot (1 - \text{bigflood}[y] \cdot SM4) \cdot (1 - \text{drought}[y] \cdot DM4) \\
GM5 \leftarrow aM5 \cdot bM5 \cdot (1 - \text{bigflood}[y] \cdot SM5) \cdot (1 - \text{drought}[y] \cdot DM5) \\
PM5 \leftarrow (1 - aM5) \cdot (1 - \text{bigflood}[y] \cdot SM5) \cdot (1 - \text{drought}[y] \cdot DM5) \\
PM6 \leftarrow (1 - aM6) \cdot (1 - \text{bigflood}[y] \cdot SM6) \cdot (1 - \text{drought}[y] \cdot DM6)

# FECUNDITY ------------------------------------------

# Assumes that if any breeding tam or cot is present, they will seed all recently-scoured substrates, although success is scaled by the lognormal hydrograph drawdown functions. # If cot and tam seedlings behave entirely independently as modeled here, this means that # under the right conditions there can be overseeding such that if (K-occupied) is the # total amount of scoured habitat remaining, then up to 2*(K-occupied) could be colonized # by seedlings, half tam and half cot. One consequence of this is if an extended drought # is followed by a good flood and then many growth years, populations could exceed K for # many years, and when pop > K, recruitment will not occur (although flood-related # mortality will keep lowering population sizes).

# Post flood space occupied ------------------------------------------

# postfloodC gives the amount of space occupied by cottonwood after the flood (AND # ACTUALLY AFTER DROUGHT OR NORMAL YEARS AS WELL!!) IN COTTONWOOD SEEDLING UNITS
postfloodC \leftarrow
\[
\begin{align*}
\text{NC}[1] & \cdot \frac{GC1}{bC1} + \\
\text{NC}[2] & \cdot \frac{GC2}{bC2 \cdot bC1} + \\
\text{NC}[3] & \cdot \frac{GC3}{bC3 \cdot bC2 \cdot bC1} + \\
\text{NC}[4] & \cdot \frac{GC4}{bC4 \cdot bC3 \cdot bC2 \cdot bC1} + \\
\text{NC}[5] & \cdot \frac{PC5}{bC4 \cdot bC3 \cdot bC2 \cdot bC1} + \\
\text{NC}[6] & \cdot \frac{PC6}{bC5 \cdot bC4 \cdot bC3 \cdot bC2 \cdot bC1}
\end{align*}
\]

# postfloodT gives the amount of space occupied by tamarisk after the flood
# IN TAMARISK SEEDLING UNITS
postfloodT <-
\[
\begin{align*}
\text{NT}[1] & \cdot \frac{GT1}{bT1} + \\
\text{NT}[2] & \cdot \frac{GT2}{bT2 \cdot bT1} + \\
\text{NT}[3] & \cdot \frac{GT3}{bT3 \cdot bT2 \cdot bT1} + \\
\text{NT}[3] & \cdot \frac{PT3}{bT2 \cdot bT1} + \\
\text{NT}[4] & \cdot \frac{GT4}{bT4 \cdot bT3 \cdot bT2 \cdot bT1} + \\
\text{NT}[4] & \cdot \frac{PT4}{bT3 \cdot bT2 \cdot bT1} + \\
\text{NT}[5] & \cdot \frac{GT5}{bT5 \cdot bT4 \cdot bT3 \cdot bT2 \cdot bT1} + \\
\text{NT}[5] & \cdot \frac{PT5}{bT4 \cdot bT3 \cdot bT2 \cdot bT1} + \\
\text{NT}[6] & \cdot \frac{PT6}{bT5 \cdot bT4 \cdot bT3 \cdot bT2 \cdot bT1}
\end{align*}
\]

# postfloodW gives the amount of space occupied by willow after the flood
# IN WILLOW SEEDLING UNITS
postfloodW <-
\[
\begin{align*}
\text{NW}[1] & \cdot \frac{GW1}{bW1} + \\
\text{NW}[2] & \cdot \frac{GW2}{bW2 \cdot bW1} + \\
\text{NW}[3] & \cdot \frac{GW3}{bW3 \cdot bW2 \cdot bW1} + \\
\text{NW}[4] & \cdot \frac{GW4}{bW4 \cdot bW3 \cdot bW2 \cdot bW1} + \\
\text{NW}[5] & \cdot \frac{PW5}{bW4 \cdot bW3 \cdot bW2 \cdot bW1} + \\
\text{NW}[5] & \cdot \frac{GW5}{bW5 \cdot bW4 \cdot bW3 \cdot bW2 \cdot bW1} + \\
\text{NW}[6] & \cdot \frac{PW6}{bW5 \cdot bW4 \cdot bW3 \cdot bW2 \cdot bW1}
\end{align*}
\]

# postfloodS gives the amount of space occupied by SAGEBRUSH after the flood
# IN SAGEBRUSH SEEDLING UNITS
postfloodS <-
\[
\begin{align*}
\text{NS}[1] & \cdot \frac{GS1}{bS1} + \\
\text{NS}[2] & \cdot \frac{GS2}{bS2 \cdot bS1} + \\
\text{NS}[3] & \cdot \frac{GS3}{bS3 \cdot bS2 \cdot bS1} + \\
\text{NS}[4] & \cdot \frac{GS4}{bS4 \cdot bS3 \cdot bS2 \cdot bS1} + \\
\text{NS}[5] & \cdot \frac{PS5}{bS4 \cdot bS3 \cdot bS2 \cdot bS1} + \\
\text{NS}[6] & \cdot \frac{PS6}{bS4 \cdot bS3 \cdot bS2 \cdot bS1}
\end{align*}
\]
NS[5] * PS5/(bS4 * bS3 * bS2 * bS1) +
NS[5] * GS5/(bS5 * bS4 * bS3 * bS2 * bS1) +
NS[6] * PS6/(bS5 * bS4 * bS3 * bS2 * bS1)

# postfloodM gives the amount of space occupied by meadow after the flood
# IN MEADOW SEEDLING UNITS
postfloodM <-
    NM[1] * GM1/bM1 +
    NM[2] * GM2/(bM2 * bM1) +
    NM[3] * GM3/(bM3 * bM2 * bM1) +
    NM[4] * GM4/(bM4 * bM3 * bM2 * bM1) +
    NM[5] * PM5/(bM4 * bM3 * bM2 * bM1) +
    NM[5] * GM5/(bM5 * bM4 * bM3 * bM2 * bM1) +
    NM[6] * PM6/(bM5 * bM4 * bM3 * bM2 * bM1)

# POTENTIAL COTTONWOOD FECUNDITY -----------------------------------------------
-------------------------------------
FC6 <- checkpos((adult_func(NC[6])) * # checks to see if at least 1 adult is present
    (1/nonind(NC[6])) *
    bigflood[y] *
    cproportion[y] *
    decline[y] *
    (K - (postfloodC +
        postfloodT * (denC1/denT1) +
        postfloodW * (denC1/denW1) +
        postfloodS * (denC1/denS1) +
        postfloodM * (denC1/denM1))))

# '(1/nonind(NC[6]))' keeps FC6 from dividing by zero by substituting an arbitrary non-0
# number that will be multiplied by 0 later anyway during matrix multiplication

# This gives POTENTIAL MAX fecundity based on amount of bare substrate available AFTER
# that year's flood.
# Reproduction is conditional on:
# 1. at least one reproductive cottonwood being present
# 2. a big flood occurring
# 3. flood during seedset window.
# It is independent of # of repro adults, but it is scaled by the rate of flooddecline AND
# BY THE PROPORTION OF SEEDSET DAYS.
# Number of new seedlings is determined by the total amount of bare substrates;
# i.e. whatever is not occupied by surviving cottonwood OR OTHER SPECIES.
# The denC1/denT1 term converts tam to cot seedling units and likewise for other 3 guilds.
# The 1/nonind[NC[6]] term will cancel out with NC[6] during matrix projection.
# i.e. number of seedlings is independent of number of mature trees.

# POTENTIAL TAMARISK FECUNDITY ----------------------------------------

    bigflood[y] *
    tproportion[y] *
    tdecline[y] *
    (denT1/denC1) *
    (K - (postfloodC +
        postfloodT * (denC1/denT1) +
        postfloodW * (denC1/denW1) +
        postfloodS * (denC1/denS1) +
        postfloodM * (denC1/denM1))))

# This gives fecundities of TAMARISK seedlings based on amount of bare substrates
# available AFTER that year's flood.
# Note that units are cot seedlings in the (K-occupied) term, but are then converted back
# to tam seedlings.
# Repro is conditional on
# 1. at least one repro tamarisk being present,
# 2. a big flood occurring
# 3. during seedset.
# It is independent of # of repro adults, but it is scaled by the rate of flooddecline and
# by prop of seedset days.
# Note that the independence of FT from breeding pop size is achieved differently here
# than for cot, it is "forced" during matrix iteration, below.
# WILLOW FECUNDITY

\[ \text{bigflood}[y] \times \]
\[ (\text{denW1/denC1}) \times \]
\[ (K - (\text{postfloodC} + \]
\[ \text{postfloodT} \times (\text{denC1/denT1}) + \]
\[ \text{postfloodW} \times (\text{denC1/denW1}) + \]
\[ \text{postfloodS} \times (\text{denC1/denS1}) + \]
\[ \text{postfloodM} \times (\text{denC1/denM1}))) \]

# checks to make sure at least one stage 3 to 6 age individual is present, and that a flood occurs. No dependence on drawdown or seedset timing.

# SAGEBRUSH FECUNDITY

\[ \text{nonflood}[y] \times \]
\[ (\text{denS1/denC1}) \times \]
\[ (K - (\text{postfloodC} + \]
\[ \text{postfloodT} \times (\text{denC1/denT1}) + \]
\[ \text{postfloodW} \times (\text{denC1/denW1}) + \]
\[ \text{postfloodS} \times (\text{denC1/denS1}) + \]
\[ \text{postfloodM} \times (\text{denC1/denM1}))) \]

# Sagebrush fecundity in # of sagebrush seedlings, to be added using "placeholder" to NS1 during iteration.
# Here, sagebrush can colonize any empty portion of K, but only during NONFLOOD years.

# MEADOW FECUNDITY

\[ \text{bigflood}[y] \times \]
\[ (\text{denM1/denC1}) \times \]
\[ (K - (\text{postfloodC} + \]
\[ \text{postfloodT} \times (\text{denC1/denT1}) + \]
\[ \text{postfloodW} \times (\text{denC1/denW1}) + \]
\[ \text{postfloodS} \times (\text{denC1/denS1}) + \]
\[ \text{postfloodM} \times (\text{denC1/denM1}))) \]
# Meadow fecundity in # of seedlings, to be added using "placeholder" to NM1 during iter.
# Here, meadow can colonize any empty portion of K.

# K

# Cottonwood
# gives total cottonwood population size as a percentage of K;
# this is the total space occupied by this species in cottonwood seedling units
KC <- 100 * (NC[1] + NC[2]/(bC1) + NC[3]/(bC2 * bC1) + NC[4]/(bC3 * bC2 * bC1) + NC[5]/(bC4 * bC3 * bC2 * bC1) + NC[6]/(bC5 * bC4 * bC3 * bC2 * bC1))/K

# same as above, but without seedlings
CnonseedK <- 100 * (NC[3]/(bC1) + NC[4]/(bC2 * bC1) + NC[5]/(bC3 * bC2 * bC1) + NC[6]/(bC4 * bC3 * bC2 * bC1) + NC[6]/(bC5 * bC4 * bC3 * bC2 * bC1))/K

# Tamarisk
KT <- 100 * (denC1/denT1) * (NT[1] + NT[2]/(bT1) + NT[3]/(bT2 * bT1) + NT[4]/(bT3 * bT2 * bT1) + NT[5]/(bT4 * bT3 * bT2 * bT1) + NT[6]/(bT5 * bT4 * bT3 * bT2 * bT1))/K

TnonseedK <- 100 * (denC1/denT1) * (NT[2]/(bT1) + NT[3]/(bT2 * bT1) + NT[4]/(bT3 * bT2 * bT1) + NT[5]/(bT4 * bT3 * bT2 * bT1) + NT[6]/(bT5 * bT4 * bT3 * bT2 * bT1))/K

# Willow
KW <- 100 * (NW[1] + NW[2]/(bW1) + NW[3]/(bW2 * bW1) + NW[4]/(bW3 * bW2 * bW1) +
\[ \text{WnonseedK} \leftarrow 100 \cdot \frac{\text{NW}[2]}{\text{bW1}} + \frac{\text{NW}[3]}{\text{bW2 \cdot bW1}} + \frac{\text{NW}[4]}{\text{bW3 \cdot bW2 \cdot bW1}} + \frac{\text{NW}[5]}{\text{bW4 \cdot bW3 \cdot bW2 \cdot bW1}} + \frac{\text{NW}[6]}{\text{bW5 \cdot bW4 \cdot bW3 \cdot bW2 \cdot bW1}} / K \]

# Sagebrush
\[ \text{KS} \leftarrow 100 \cdot \frac{\text{NS}[1]}{\text{bS1}} + \frac{\text{NS}[2]}{\text{bS2 \cdot bS1}} + \frac{\text{NS}[3]}{\text{bS3 \cdot bS2 \cdot bS1}} + \frac{\text{NS}[4]}{\text{bS4 \cdot bS3 \cdot bS2 \cdot bS1}} + \frac{\text{NS}[5]}{\text{bS5 \cdot bS4 \cdot bS3 \cdot bS2 \cdot bS1}} / K \]

# Meadow
\[ \text{KM} \leftarrow 100 \cdot \frac{\text{NM}[1]}{\text{bM1}} + \frac{\text{NM}[2]}{\text{bM2 \cdot bM1}} + \frac{\text{NM}[3]}{\text{bM3 \cdot bM2 \cdot bM1}} + \frac{\text{NM}[4]}{\text{bM4 \cdot bM3 \cdot bM2 \cdot bM1}} + \frac{\text{NM}[5]}{\text{bM5 \cdot bM4 \cdot bM3 \cdot bM2 \cdot bM1}} / K \]

# TRANSITION MATRICES

# TRANSITION MATRIX FOR cottonwood
\[ \text{AC1} \leftarrow c(0, 0, 0, 0, 0, \text{FC6}) \]
\[ \text{AC2} \leftarrow c(\text{GC1}, 0, 0, 0, 0, 0) \]
AC3 <- c(0, GC2, 0, 0, 0, 0)
AC4 <- c(0, 0, GC3, 0, 0, 0)
AC5 <- c(0, 0, 0, GC4, PC5, 0)
AC6 <- c(0, 0, 0, GC5, PC6)
# Matrix
AC <- rbind(AC1, AC2, AC3, AC4, AC5, AC6)

# TRANSITION MATRIX FOR tamarisk

# Note: fecundity is not included here, since it is assigned directly
during iteration
AT1 <- c(0, 0, 0, 0, 0, 0)
AT2 <- c(GT1, 0, 0, 0, 0, 0)
AT3 <- c(0, GT2, PT3, 0, 0, 0)
AT4 <- c(0, 0, GT3, PT4, 0, 0)
AT5 <- c(0, 0, 0, GT4, PT5, 0)
AT6 <- c(0, 0, 0, 0, GT5, PT6)
# Matrix
AT <- rbind(AT1, AT2, AT3, AT4, AT5, AT6)

# TRANSITION MATRIX FOR dynamic riverbank specialist, willow

# Similar stage structure to cot, except that reproduction can occur
in all but 1st yr
AW1 <- c(0, 0, 0, 0, 0, 0)
AW2 <- c(GW1, 0, 0, 0, 0, 0)
AW3 <- c(0, GW2, 0, 0, 0, 0)
AW4 <- c(0, 0, GW3, 0, 0, 0)
AW5 <- c(0, 0, 0, GW4, PW5, 0)
AW6 <- c(0, 0, 0, GW5, PW6)
# Matrix
AW <- rbind(AW1, AW2, AW3, AW4, AW5, AW6)

# TRANSITION MATRIX FOR sagebrush

# Arid shrubland indicator big sagebrush.
# Similar stage structure to cottonwood, fecundity is assigned
directly as with tamarisk
# during iteration.
AS1 <- c(0, 0, 0, 0, 0)
AS2 <- c(GS1, 0, 0, 0, 0)
AS3 <- c(0, GS2, 0, 0, 0)
AS4 <- c(0, 0, GS3, 0, 0, 0)
AS5 <- c(0, 0, 0, GS4, 0, 0)
AS6 <- c(0, 0, 0, GS5, GS6)
# Matrix
AS <- rbind(AS1, AS2, AS3, AS4, AS5, AS6)

# TRANSITION MATRIX FOR 5th species, xeric meadow
# Includes grasses such as wheatgrass.
# Similar stage structure to cottonwood, fecundity is assigned
# directly as with tamarisk during iteration.
AM1 <- c(0, 0, 0, 0, 0, 0)
AM2 <- c(GM1, 0, 0, 0, 0, 0)
AM3 <- c(0, GM2, 0, 0, 0, 0)
AM4 <- c(0, 0, GM3, 0, 0, 0)
AM5 <- c(0, 0, 0, GM4, 0, 0)
AM6 <- c(0, 0, 0, 0, GM5, PM6)
# Matrix
AM <- rbind(AM1, AM2, AM3, AM4, AM5, AM6)

# COMPILING OUTPUTS
# Cottonwood
Coutput[i,1:6] <- log(NC + 1) # array of no. ind. of each age class for each yr
# projected. NC = total no. ind. for each age class
DomC[i] <- NC[5] # vector of cot age class 5 for each year projected
Cnonseedling[i] <- CnonseedK # total cottonwood pop. size as % of K without seedlings for
# each projected year
Cspaceoutput[i] <- KC # total cottonwood pop. size as % of K; this is the total space
# occupied by this species in cottonwood seedling units

# Tamarisk - same as cottonwood
Toutput[i,1:6] <- log(NT + 1)
Tnonseedling[i] <- TnonseedK
Tspaceoutput[i] <- KT

# Willow
Woutput[i,1:6] <- log(NW + 1)
Wnonseedling[i] <- WnonseedK
Wspaceoutput[i] <- KW

# Sagebrush
Soutput[i,1:6] <- log(NS + 1)
Snonseedling[i] <- SnonseedK
Sspaceoutput[i] <- KS

# Meadow
Moutput[i,1:6] <- log(NM + 1)
Mnonseedling[i] <- MnonseedK
Mspaceoutput[i] <- KM

# Records flood settings of each particular projected year (0 for nonflood, 1 for flood)
floodoutput[i] <- bigflood[y]

droughtoutput[i] <- drought[y]

# Same for nonflood
nonfloodoutput[i] <- nonflood[y]

# Same for normal
normaloutput[i] <- ifelse(bigflood[y] == 0 & drought[y] == 0 & nonflood[y] == 1, 1, 0)

# Fecundity of all but cottonwood to put into matrix projection.
# Cottonwood is already in matrix
Tplaceholder <- FT
Splaceholder <- FS
Wplaceholder <- FW
Mplaceholder <- FW

# MATRIX MULTIPLICATION
# Cottonwood
NC <- AC %*% NC # AC is transition matrix, NC = total no. ind. for each age class
NC <- quasi(NC) # quasi extinction threshold of 1: below 1 go to 0
# Tamarisk
# Note the use of 'placeholders' for fecundity in the following guilds
NT <- AT %*% NT
NT[1] <- Tplaceholder
NT <- quasi(NT)

# Willow
NW <- AW %*% NW
NW[1] <- Wplaceholder
NW <- quasi(NW)

# Sagebrush
NS <- AS %*% NS
NS[1] <- Splaceholder
NS <- quasiten(NS)

# Meadow
NM <- AM %*% NM
NM[1] <- Mplaceholder
NM <- quasi(NM)
} # End of inner loop

# Mean vals for each replicate run over period specified from burning to end of projection
# No seedlings
Crep[rep] <- mean(Cnonseedling[seq(burnin + 1, count)])
Trep[rep] <- mean(Tnonseedling[seq(burnin + 1, count)])
Wrep[rep] <- mean(Wnonseedling[seq(burnin + 1, count)])
Srep[rep] <- mean(Snonseedling[seq(burnin + 1, count)])
Mrep[rep] <- mean(Mnonseedling[seq(burnin + 1, count)])

# All incl. seedlings
Crep_all[rep] <- mean(Cspaceoutput[seq(burnin + 1, count)])
Trep_all[rep] <- mean(Tspaceoutput[seq(burnin + 1, count)])
Wrep_all[rep] <- mean(Wspaceoutput[seq(burnin + 1, count)])
Srep_all[rep] <- mean(Sspaceoutput[seq(burnin + 1, count)])
Mrep_all[rep] <- mean(Mspaceoutput[seq(burnin + 1, count)])

# Stage 5 cot, stage 4 tam - as per Merritt and Poff 2010
DomCrep[rep] <- mean(DomC[seq(burnin + 1, count)])
DomTrep[rep] <- mean(DomT[seq(burnin + 1, count)])
} # End of mid loop

# Outer loop compilation - results of flow mod scenarios. Not useful unless simulating
# changes to flow regime
# Adults - mean of Crep (Mean values for each replicate run over the period specified from
# burnin to end of projection)
# This is then the mean of each flow modification setting for the full burnin -> end of
# projection period.

# No seedlings
Cgraph[zim] <- mean(Crep)
Tgraph[zim] <- mean(Trep)
Wgraph[zim] <- mean(Wrep)
Sgraph[zim] <- mean(Srep)
Mgraph[zim] <- mean(Mrep)

# All incl. seedlings
Cgraph_all[zim] <- mean(Crep_all)
Tgraph_all[zim] <- mean(Trep_all)
Wgraph_all[zim] <- mean(Wrep_all)
Sgraph_all[zim] <- mean(Srep_all)
Mgraph_all[zim] <- mean(Mrep_all)

# Stage 5 cot, stage 4 tam - as per Merritt and Poff 2010
DomCrep_graph[zim] <- mean(DomCrep)
DomTrep_graph[zim] <- mean(DomTrep)

# Proportion of drought years in model run
droughtgraph[zim] <- sum(drought)/length(bigflood)

# Proportion of flood years in model run
floodgraph[zim] <- sum(bigflood)/length(bigflood)

# Proportion of nonflood years in model run
nonfloodgraph[zim] <- sum(nonflood)/length(bigflood)
# Proportion of normal years in model run (not floods and not droughts)
normalgraph[zim] <-
    sum(ifelse(bigflood == 0 & drought == 0 & nonflood == 1, 1, 0))/length(bigflood)

# Adding 1 to droughtchanged and floodchanged - this keeps going until reaching the number
# specified in outerreps (i.e. 84 mods all years)
droughtchanged = floodchanged = droughtchanged + 1

} # End outer loop

# This will change depending on what simulations we are running
# Showing a couple of example outputs below. One for flow mod scenario and one just
# showing the core results for the projection period

# Flow mod graph
# only relevant if OUTER LOOP is run with flow modification scenarios
# Showing here just the graph of how drought scenarios affect the 5 guilds including all
# stages, incl. seedlings.

# Compiling all results from flow mods into a dataframe
graph_df <- as.data.frame(cbind(Cgraph, Tgraph, Wgraph, Sgraph, Mgraph, droughtgraph, 
    floodgraph, nonfloodgraph, 
    normalgraph, Cgraph_all,
Tgraph_all, Wgraph_all, Sgraph_all,
Mgraph_all))

# Adding a column of replicate number
graph_df$replicate <- as.numeric(as.character(row.names(graph_df)))

# Gathering dataframe for plotting
graph_df_g <- gather(graph_df, key, value, Cgraph:Mgraph) # WITHOUT seedlings
graph_df_g_all <- gather(graph_df, key, value, Cgraph_all:Mgraph_all) # WITH seedlings

# 5 guilds mean percent of K over full drought mod settings during the burnin to full
# projection period (incl. seedlings)
# uncomment if running flow mod scenario
# droughtplot_all <- ggplot(graph_df_g_all,
# aes(droughtgraph, value, colour = key,
#     linetype = key)) +
#     geom_path() +
#     theme_classic() +
#     scale_colour_brewer(type = 'qual', palette = 6)
# droughtplot_all

# Space occupied in each year over full projection -----------------------
# Results from MIDDLE LOOP - replications of each setting

# All incl. seedlings
space_df <- as.data.frame(cbind(Cspaceoutput, Tspaceoutput, 
Wspaceoutput, Sspaceoutput, Mspaceoutput))

# Adding a year column
space_df$year <- as.numeric(as.character(row.names(space_df)))

# Gathering into long form for plotting
space_df_g <- gather(space_df, key, value, Cspaceoutput:Mspaceoutput)

# Plotting
space_plot <- ggplot(space_df_g, aes(year, value, colour = key)) + geom_path()
space_plot
# OVERVIEW OF PROCESS

# 1. draw a year at random from hydrograph
# 2. log the current values of the five guilds, store these in a vector for output
# 3. calculate space occupied by each guild as a percentage of K
# 4. keep track of whether a drought or flood year
# 6. store non-cottonwood guild fecundities in placeholder. At this point FT/FW/FS/FM look
# ahead and calculate how much space will be left after cot and tam flood mortality
# occurs. Had to be done before matrix projection because FT/FW/FS/FM depend on current
# matrix values of all guilds.
# 7. Project populations. The quasi function turns numbers less than one to zero. quasiten
# is used for sagebrush (<10 = 0).

# VARIOUS CHECKS OF FINAL RUN

floodfreq <- sum(bigflood)/length(bigflood)
droughtfreq <- sum(drought)/length(bigflood)
normfreq <- (length(bigflood) - (sum(drought) + sum(bigflood)))/length(bigflood)
sum(floodfreq + droughtfreq + normfreq)

defoldchanged
defoldfreq
droughtfreq
normfreq

# END

Sample hydrograph input data for riparian model (Maybell.csv):
[link to data](https://figshare.com/articles/5-guild_riparian_flow-population_model/4652608)

<p>| year | floodmag | floodates | slopecot | slopetam | basedates |</p>
<table>
<thead>
<tr>
<th>Year</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1916</td>
<td>11700</td>
<td>223</td>
<td>-2.93</td>
<td>-1.28</td>
<td>295</td>
</tr>
<tr>
<td>1917</td>
<td>17900</td>
<td>231</td>
<td>-2.33</td>
<td>-3.42</td>
<td>315</td>
</tr>
<tr>
<td>1918</td>
<td>10500</td>
<td>258</td>
<td>-1.88</td>
<td>-2.4</td>
<td>306</td>
</tr>
<tr>
<td>1919</td>
<td>7670</td>
<td>233</td>
<td>-3.18</td>
<td>-0.68</td>
<td>279</td>
</tr>
<tr>
<td>1920</td>
<td>16000</td>
<td>239</td>
<td>-4.62</td>
<td>-1.95</td>
<td>307</td>
</tr>
<tr>
<td>1921</td>
<td>17700</td>
<td>259</td>
<td>-4.39</td>
<td>-1.71</td>
<td>318</td>
</tr>
<tr>
<td>1922</td>
<td>10800</td>
<td>240</td>
<td>-3.14</td>
<td>-1.33</td>
<td>293</td>
</tr>
<tr>
<td>1923</td>
<td>10900</td>
<td>240</td>
<td>-2.92</td>
<td>-1.62</td>
<td>310</td>
</tr>
<tr>
<td>1924</td>
<td>7810</td>
<td>235</td>
<td>-3.18</td>
<td>-0.68</td>
<td>291</td>
</tr>
<tr>
<td>1925</td>
<td>6640</td>
<td>242</td>
<td>-3.03</td>
<td>-0.82</td>
<td>292</td>
</tr>
<tr>
<td>1926</td>
<td>11800</td>
<td>231</td>
<td>-2.16</td>
<td>-1.89</td>
<td>294</td>
</tr>
<tr>
<td>1927</td>
<td>13700</td>
<td>225</td>
<td>-4.26</td>
<td>-1.45</td>
<td>300</td>
</tr>
<tr>
<td>1928</td>
<td>14400</td>
<td>226</td>
<td>-2.34</td>
<td>-2.21</td>
<td>319</td>
</tr>
<tr>
<td>1929</td>
<td>7980</td>
<td>244</td>
<td>-3.53</td>
<td>-0.57</td>
<td>279</td>
</tr>
<tr>
<td>1930</td>
<td>6500</td>
<td>231</td>
<td>-2.15</td>
<td>-0.81</td>
<td>279</td>
</tr>
<tr>
<td>1931</td>
<td>12100</td>
<td>236</td>
<td>-1.97</td>
<td>-1.94</td>
<td>316</td>
</tr>
<tr>
<td>1932</td>
<td>11200</td>
<td>246</td>
<td>-4.12</td>
<td>-1.35</td>
<td>288</td>
</tr>
<tr>
<td>1933</td>
<td>4080</td>
<td>223</td>
<td>-2.36</td>
<td>-0.05</td>
<td>253</td>
</tr>
<tr>
<td>1934</td>
<td>9870</td>
<td>259</td>
<td>-2.87</td>
<td>-1.73</td>
<td>291</td>
</tr>
<tr>
<td>1935</td>
<td>10600</td>
<td>230</td>
<td>-3.36</td>
<td>-1.14</td>
<td>283</td>
</tr>
<tr>
<td>1936</td>
<td>10000</td>
<td>229</td>
<td>-1.86</td>
<td>-0.97</td>
<td>297</td>
</tr>
<tr>
<td>1937</td>
<td>12100</td>
<td>231</td>
<td>-3.69</td>
<td>-1.57</td>
<td>295</td>
</tr>
<tr>
<td>1938</td>
<td>7860</td>
<td>219</td>
<td>-3.28</td>
<td>-0.92</td>
<td>280</td>
</tr>
<tr>
<td>1939</td>
<td>9170</td>
<td>226</td>
<td>-3.48</td>
<td>-0.72</td>
<td>280</td>
</tr>
<tr>
<td>1940</td>
<td>11700</td>
<td>227</td>
<td>-2.76</td>
<td>-1</td>
<td>285</td>
</tr>
<tr>
<td>1941</td>
<td>9930</td>
<td>239</td>
<td>-3.67</td>
<td>-1.21</td>
<td>288</td>
</tr>
<tr>
<td>1942</td>
<td>9280</td>
<td>246</td>
<td>-2.92</td>
<td>-1.93</td>
<td>290</td>
</tr>
<tr>
<td>1943</td>
<td>9080</td>
<td>237</td>
<td>-3.6</td>
<td>-1.7</td>
<td>289</td>
</tr>
<tr>
<td>1944</td>
<td>10900</td>
<td>225</td>
<td>-1.81</td>
<td>-1.68</td>
<td>328</td>
</tr>
<tr>
<td>1945</td>
<td>6850</td>
<td>211</td>
<td>-1.62</td>
<td>-0.92</td>
<td>286</td>
</tr>
<tr>
<td>1946</td>
<td>12400</td>
<td>222</td>
<td>-1.89</td>
<td>-1.77</td>
<td>303</td>
</tr>
<tr>
<td>1947</td>
<td>11300</td>
<td>234</td>
<td>-3.3</td>
<td>-1.12</td>
<td>287</td>
</tr>
<tr>
<td>1948</td>
<td>9730</td>
<td>262</td>
<td>-2.8</td>
<td>-1.96</td>
<td>303</td>
</tr>
<tr>
<td>1949</td>
<td>8210</td>
<td>237</td>
<td>-2.62</td>
<td>-1.57</td>
<td>292</td>
</tr>
<tr>
<td>1950</td>
<td>8870</td>
<td>242</td>
<td>-3.02</td>
<td>-1.74</td>
<td>305</td>
</tr>
<tr>
<td>1951</td>
<td>13800</td>
<td>249</td>
<td>-3.39</td>
<td>-1.24</td>
<td>293</td>
</tr>
<tr>
<td>1952</td>
<td>10100</td>
<td>258</td>
<td>-3.34</td>
<td>-1.11</td>
<td>295</td>
</tr>
<tr>
<td>1953</td>
<td>5480</td>
<td>235</td>
<td>-1.96</td>
<td>-0.72</td>
<td>275</td>
</tr>
<tr>
<td>Year</td>
<td>Value</td>
<td>Previous Year Value</td>
<td>Change</td>
<td>Previous Year Change</td>
<td>Index</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>---------------------</td>
<td>--------</td>
<td>----------------------</td>
<td>-------</td>
</tr>
<tr>
<td>1955</td>
<td>7000</td>
<td>228</td>
<td>-2.38</td>
<td>-0.98</td>
<td>281</td>
</tr>
<tr>
<td>1956</td>
<td>9870</td>
<td>237</td>
<td>-3.64</td>
<td>-0.68</td>
<td>281</td>
</tr>
<tr>
<td>1957</td>
<td>15700</td>
<td>252</td>
<td>-1.44</td>
<td>-3.46</td>
<td>331</td>
</tr>
<tr>
<td>1958</td>
<td>12200</td>
<td>242</td>
<td>-4.7</td>
<td>-1.04</td>
<td>283</td>
</tr>
<tr>
<td>1959</td>
<td>6690</td>
<td>253</td>
<td>-2.45</td>
<td>-1.49</td>
<td>286</td>
</tr>
<tr>
<td>1960</td>
<td>8000</td>
<td>227</td>
<td>-2.76</td>
<td>-1</td>
<td>284</td>
</tr>
<tr>
<td>1961</td>
<td>6350</td>
<td>243</td>
<td>-3.28</td>
<td>-0.77</td>
<td>277</td>
</tr>
<tr>
<td>1962</td>
<td>11500</td>
<td>226</td>
<td>-1.42</td>
<td>-2.31</td>
<td>300</td>
</tr>
<tr>
<td>1963</td>
<td>6290</td>
<td>224</td>
<td>-2.29</td>
<td>-0.35</td>
<td>273</td>
</tr>
<tr>
<td>1964</td>
<td>9990</td>
<td>234</td>
<td>-2.36</td>
<td>-1.82</td>
<td>293</td>
</tr>
<tr>
<td>1965</td>
<td>11800</td>
<td>258</td>
<td>-1.9</td>
<td>-1.78</td>
<td>314</td>
</tr>
<tr>
<td>1966</td>
<td>6900</td>
<td>167</td>
<td>-2.61</td>
<td>-0.7</td>
<td>273</td>
</tr>
<tr>
<td>1967</td>
<td>8890</td>
<td>239</td>
<td>-2.51</td>
<td>-1.85</td>
<td>301</td>
</tr>
<tr>
<td>1968</td>
<td>11400</td>
<td>250</td>
<td>-3.34</td>
<td>-1.72</td>
<td>296</td>
</tr>
<tr>
<td>1969</td>
<td>8290</td>
<td>208</td>
<td>-2.44</td>
<td>-1.4</td>
<td>299</td>
</tr>
<tr>
<td>1970</td>
<td>12700</td>
<td>235</td>
<td>-3.17</td>
<td>-2.3</td>
<td>305</td>
</tr>
<tr>
<td>1971</td>
<td>10300</td>
<td>244</td>
<td>-3.32</td>
<td>-2.14</td>
<td>303</td>
</tr>
<tr>
<td>1972</td>
<td>8890</td>
<td>253</td>
<td>-3.15</td>
<td>-1</td>
<td>283</td>
</tr>
<tr>
<td>1973</td>
<td>12100</td>
<td>234</td>
<td>-1.41</td>
<td>-2.09</td>
<td>308</td>
</tr>
<tr>
<td>1974</td>
<td>15400</td>
<td>223</td>
<td>-3.48</td>
<td>-1.68</td>
<td>302</td>
</tr>
<tr>
<td>1975</td>
<td>11700</td>
<td>252</td>
<td>-1.05</td>
<td>-2.38</td>
<td>309</td>
</tr>
<tr>
<td>1976</td>
<td>7450</td>
<td>235</td>
<td>-2.27</td>
<td>-1.31</td>
<td>290</td>
</tr>
<tr>
<td>1977</td>
<td>3620</td>
<td>248</td>
<td>-1.98</td>
<td>0.12</td>
<td>263</td>
</tr>
<tr>
<td>1978</td>
<td>11570</td>
<td>256</td>
<td>-2.28</td>
<td>-2.62</td>
<td>309</td>
</tr>
<tr>
<td>1979</td>
<td>13600</td>
<td>242</td>
<td>-3.11</td>
<td>-2.06</td>
<td>304</td>
</tr>
<tr>
<td>1980</td>
<td>11700</td>
<td>236</td>
<td>-3.05</td>
<td>-1.6</td>
<td>292</td>
</tr>
<tr>
<td>1981</td>
<td>6490</td>
<td>252</td>
<td>-2.7</td>
<td>-0.94</td>
<td>270</td>
</tr>
<tr>
<td>1982</td>
<td>10700</td>
<td>217</td>
<td>-1.99</td>
<td>-2.48</td>
<td>312</td>
</tr>
<tr>
<td>1983</td>
<td>13400</td>
<td>242</td>
<td>-2.82</td>
<td>-2.8</td>
<td>329</td>
</tr>
<tr>
<td>1984</td>
<td>25100</td>
<td>229</td>
<td>-3.57</td>
<td>-2.41</td>
<td>319</td>
</tr>
<tr>
<td>1985</td>
<td>13600</td>
<td>219</td>
<td>-3.16</td>
<td>-1.22</td>
<td>311</td>
</tr>
<tr>
<td>1986</td>
<td>11000</td>
<td>251</td>
<td>-2.92</td>
<td>-1.86</td>
<td>309</td>
</tr>
<tr>
<td>1987</td>
<td>6140</td>
<td>230</td>
<td>-1.29</td>
<td>-0.44</td>
<td>269</td>
</tr>
<tr>
<td>1988</td>
<td>10200</td>
<td>231</td>
<td>-2.79</td>
<td>-1.41</td>
<td>284</td>
</tr>
<tr>
<td>1989</td>
<td>4940</td>
<td>225</td>
<td>-2.44</td>
<td>-0.69</td>
<td>273</td>
</tr>
<tr>
<td>1990</td>
<td>7020</td>
<td>256</td>
<td>-2.23</td>
<td>-1.23</td>
<td>286</td>
</tr>
<tr>
<td>1991</td>
<td>8560</td>
<td>235</td>
<td>-2.75</td>
<td>-0.81</td>
<td>280</td>
</tr>
<tr>
<td>1992</td>
<td>5900</td>
<td>240</td>
<td>-1.59</td>
<td>-0.52</td>
<td>273</td>
</tr>
<tr>
<td>1993</td>
<td>12800</td>
<td>234</td>
<td>-3.03</td>
<td>-2.17</td>
<td>303</td>
</tr>
<tr>
<td>Year</td>
<td>Value</td>
<td>Rate</td>
<td>Percentage</td>
<td>Value</td>
<td>Rate</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>------------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>1994</td>
<td>5880</td>
<td>231</td>
<td>-3.16</td>
<td>0.62</td>
<td>271</td>
</tr>
<tr>
<td>1995</td>
<td>13300</td>
<td>260</td>
<td>-0.88</td>
<td>2.63</td>
<td>316</td>
</tr>
<tr>
<td>1996</td>
<td>15000</td>
<td>231</td>
<td>-2.69</td>
<td>1.84</td>
<td>296</td>
</tr>
<tr>
<td>1997</td>
<td>18800</td>
<td>247</td>
<td>-4.33</td>
<td>1.69</td>
<td>326</td>
</tr>
<tr>
<td>1998</td>
<td>10700</td>
<td>234</td>
<td>-2.71</td>
<td>1.97</td>
<td>312</td>
</tr>
</tbody>
</table>
Fish population model

Primary publication describing the fish model methodology:
https://doi.org/10.1002/ecs2.2681

Fish model vital rate estimation:

Desert sucker
- Average adult length, length-weight (Gibson et al. 2015 unpublished data using 143 individuals)
- Longevity (for congeners; Klein et al. 2017)
- GSI, ages and lengths at life stages (for congeners; Carothers and Minckley 1980, McCall 1980, McAda and Wydoski 1983, Propst et al. 2001)

Sonora sucker
- Reproductive timing/temperature, lengths (Minckley 1973, Minckley and Marsh 2009)
- Age at maturity (Frimpong and Angermeier 2009)
- Length-weight (Gibson et al. 2015, unpublished data using 109 individuals)
- Longevity (for congener; Klein et al. 2017)

Roundtail chub
- Reproductive timing/temperature, clutch size, GSI, ages and lengths at life stages, longevity (Brouder et al. 2000, Brouder 2005, Brouder et al. 2006)
- Effects of floods on recruitment (Brouder 2001)
Yellow bullhead
- Reproductive timing/temperature, GSI, lengths and ages at life stages (for congener; Copp et al. 2016)
- Length-weight (Gibson et al. 2015, unpublished data using 22 individuals)
- Age, length, and longevity (Murie et al. 2009)

Green sunfish
- Reproductive timing/temperature, GSI (Kaya and Hasler 1972)
- Clutch size (Carlander 1977)
- Length-weight (Mannes and Jester 1980)
- Age at maturity (Moyle 2002, Wang 1986)
- Lengths at life stages, longevity (Delp et al. 2000, Quist and Guy 2001)
- Resilience to drought (Bêche et al. 2009)

Smallmouth bass
- Longevity (Smith et al. 2005)
- Length-weight relationship (Lawrence et al. 2015)

Red shiner
- Length-weight (Franssen et al. 2007)
Information tables about vital rates and references for vital rates

Appendix Tables

Table S1. Flow modifiers derived from flow-ecology relationships in literature (see also Table 2 and main text for references and details). Flow modifiers ($Y_{ijk}$) act on baseline mortality rates ($M_{ij}$; Table A3) specific for stage $i$, species $j$ and flow-event year $k$. Values greater than 1 increase mortality, values less than 1 decrease mortality. Abbreviations for flow event year (SP_HF = spring high flood, SU_HF = summer high flood, SP_MF = spring medium flood, NE = non-event, DR = drought). S1 = post-larval young-of-year fishes, S2 = size at first maturity, S3 = average adult size in population.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Flow Event</th>
<th>Desert sucker</th>
<th>Sonora sucker</th>
<th>Roundtail chub</th>
<th>Yellow bullhead</th>
<th>Red shiner</th>
<th>Green sunfish</th>
<th>Smallmouth bass</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>SP_HF</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S1</td>
<td>SU_HF</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1.5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>S1</td>
<td>SP_MF</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S1</td>
<td>NE</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.3</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>S1</td>
<td>DR</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>S2, S3</td>
<td>SP_HF</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S2, S3</td>
<td>SU_HF</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S2, S3</td>
<td>SP_MF</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S2, S3</td>
<td>NE</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.3</td>
<td>0.2</td>
<td>1</td>
</tr>
<tr>
<td>S2, S3</td>
<td>DR</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>0.2</td>
<td>1</td>
<td>1.5</td>
</tr>
</tbody>
</table>
Table S2. Length-weight relationships. Length-weight relationships used to calculate biomass for individuals at all three life stages based on their lengths at the end of their first year (L₁), length at maturity (L₂) and average adult length (L₃; see also ”References for vital rates” above).

<table>
<thead>
<tr>
<th></th>
<th>Length-Weight (W = aLᵇ)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a</td>
<td>b</td>
<td>L₁</td>
<td>L₂</td>
</tr>
<tr>
<td>Desert sucker</td>
<td>9.76E-06</td>
<td>3.038</td>
<td>68</td>
<td>92</td>
</tr>
<tr>
<td>Sonora sucker</td>
<td>9.61E-06</td>
<td>3.022</td>
<td>152</td>
<td>282</td>
</tr>
<tr>
<td>Roundtail chub</td>
<td>7.89E-06</td>
<td>3.022</td>
<td>101</td>
<td>181</td>
</tr>
<tr>
<td>Yellow bullhead</td>
<td>7.03E-06</td>
<td>2.92</td>
<td>99</td>
<td>200</td>
</tr>
<tr>
<td>Red shiner</td>
<td>5.75E-06</td>
<td>3.16</td>
<td>25</td>
<td>30</td>
</tr>
<tr>
<td>Green sunfish</td>
<td>3.31E-05</td>
<td>3.356</td>
<td>45</td>
<td>65</td>
</tr>
<tr>
<td>Smallmouth bass</td>
<td>1.16E-06</td>
<td>3.02</td>
<td>70</td>
<td>200</td>
</tr>
</tbody>
</table>

Literature Cited


R code for fish model:
https://doi.org/10.5281/zenodo.1309024

## Verde Fish Model
## Jane Rogosch, Jono Tonkin, et al.
## July 2018
## Community-wide stochastic matrix population model that links population
## dynamics with river flow regimes.
## This is the bare model used in the Rogosch et al. ms.

## Required libraries
library(ggplot2)
library(plyr)
library(tidyverse)
library(popbio)

## * SETUP -----------------------------------------------
#rm(list = ls()) # clearing the workspace

## Bringing in flow data
all.scenarios.list <- readRDS('data/all_scenarios_list.rds')

## When using natural flow data, just pull it out from the list here
flowdata <- all.scenarios.list$natural.flow

count <- 54 # 54 years in flow record, if count = 45 goes to 2008
iterations <- 10 # number of replicate projections to run (mid loop)

## Modifiers
modifiers <- read.csv('data/modifiers-all-spp.csv')

## adding 'Modifier' value from csv to 'Code' in csv
for(j in 1:length(modifiers[,1])) {
  nam <- paste(modifiers[j,4])
  assign(nam, modifiers[j,5])
}
## Vital rates
## Baseline maturation probability, aCACL3 (adult senescence rate)
## Background mortality
## Initial volume in grams in 100-m reach
## Fecundity based on year type and GSI
## Stage specific densities (ind./g)

vitalrates <- read.csv('data/vital-rates.csv')

## assigning vital rate values from column 3 to 'code' in column 2
for(k in 1:length(vitalrates[,1])) {
  nam <- paste(vitalrates[k,2])
  assign(nam, vitalrates[k,3])
}

## * Key
------------------
### CACL (Catostomus clarki) – desert sucker
### GIRO (Gila robusta) – roundtail chub
### LECY (Lepomis cyanellus) – green sunfish
### CAIN (Catostomus insignis) – sonora sucker
### MIDO (Micropterus dolomieu) – smallmouth bass
### CYLU (Cyprinella lutrensis) – red shiner
### AMNA (Ameiurus natalis) – yellow bullhead

## Average total volume of water per 100 m reach in m³: 307
## Average total fish biomass per 100 m reach in g: 4766
## Average total biomass Bonar 2004 in g/100m²: 606
## Max for a 100 m reach in Gibson samples (excluding GAAF): 6996

## vector of species names
sppnames <- c('CACL', 'GIRO', 'LECY', 'CAIN', 'MIDO', 'CYLU', 'AMNA')

K = 47660 # mean for 1-km reach across 6 replicate reaches

## Loading functions from functions.R file
------------------
source('code/functions.R')

## * ITERATION PARAMETERS
------------------
## Setting up arrays/vectors to fill with data from loops
## Mid loop details

--

## 'iterations' - number of replicate flow sequences to run for averaging

years <- flowdata$year
stages <- as.character(c("S1", "S2", "S3"))

## Total. N of stages 2 and 3 each year

---

## Takes all stages 2 and 3 and sums them for each year and iteration

Total.N <- array(0,
    dim = c(54, iterations),
    dimnames = list(years, 1:iterations)
)

## replist. List of arrays w/ abundance data for each spp

---

## Creating a list of 7 arrays to fill in. One for each spp.

## Create an array to be repeated

reparray <- array(0,
    dim = c(54, 3, iterations),
    dimnames = list(years, stages, 1:iterations)
)

## Repeating the array 7 times

replist <- rep(list(reparray), 7)

## Assigning names to each array from sppnames vector

names(replist) <- sppnames

## Inner loop details

---

## 'count' - number of years to project simulations (inner loop)

## N

---

## Output of biomass and no. ind. for each age class for each year projected

## Array w/ 3 cols (stage classes) and however many rows there are yrs projected

## Creating a list of 7 arrays to fill in. One for each spp.

## Create an array to be repeated

output.N.array <- array(0, dim = c(count, 3))
## Repeating the array 7 times
output.N.list <- rep(list(output.N.array), 7)

## Assigning names to each array from sppnames vector
names(output.N.list) <- sppnames

## Create a df to fill in w/ lambda values
---------------------------
----------
lambda.df <- data.frame(matrix(ncol = 7, nrow = count))
names(lambda.df) <- sppnames

## Biomass
---------------------------
----------
## Creating a list of 7 arrays to fill in. One for each spp.
## Create an array to be repeated
output.biom.array <- array(0, dim = c(count, 3))

## Repeating the array 7 times
output.biom.list <- rep(list(output.biom.array), 7)

## Assigning names to each array from sppnames vector
names(output.biom.list) <- sppnames

## Total biomass as % of K
---------------------------
----------
## Creating a list of 7 vectors to fill in. One for each spp.
## Create a vector to be repeated
biomoutput.vector <- numeric(length = count)

## Repeating the vector 7 times
biomoutput.list <- rep(list(biomoutput.vector), 7)

## Assigning names to each vector from sppnames vector
names(biomoutput.list) <- sppnames

## Flow results
---------------------------
----------
## Flood and drought settings for each yr projected into future (i.e. 0 or 1)
## Create data frame with 5 cols and 'count' rows to fill in with flow results
flowresults <- data.frame(matrix(ncol = 5, nrow = count))
names(flowresults) <- c('SPhighflood',

## Fecundities

---

## Creating a list of 7 vectors to fill in. One for each spp.

**Create a vector to be repeated**

```r
fec.vector <- numeric(length = count)
```

**Repeating the vector 7 times**

```r
fec.list <- rep(list(fec.vector), 7)
```

**Assigning names to each vector from sppnames vector**

```r
names(fec.list) <- sppnames
```

### Middle loop

#### Middle loop uses iterator "iter" to get "iterations" for suming S2 and S3

**No 'outer' loop under normal runs.**

```r
for(iter in 1:iterations) {
  # USE THIS to examine different flow year types
  # All 2010 SPflood +
  SUnhhighflood
  # All Spflood 1993
  # All drought Y2K

  # z <- rep(47, 84)
  # z <- rep(30, 84)
  # z <- rep(37, 84)
}
```
```r
# # Nonevent 1985
# z <- rep(22, 84)

# # SUflood
# z <- rep(21, 84)

# # Medflood
# z <- rep(25, 84)

## Need to read in initial biom every time so starting biomass is reset each iteration
## N gives the total number of individuals for each age class.
## Initially here, this is found by multiplying the number of g occupied by
## a given class by the density per g
## biom = g/m3
## den = indiv/g

## To have different initial starting population sizes for each iteration,
## taking biom of stage 3 from negative binomial distribution, where the
## parameter (lambda = mean) and K (dispersion) is calculated from mean and
## variance in abundance across seven sites in Verde River from 94-08, and
## scaled to biomass from Gibson 2012 survey in file:
## "Rinne Verde River Data 1994-2008-.xlsx"

biomCACL <- c(biomCACL1,
              biomCACL2,
              rnbinom(1, size = 1.52, mu = 5284))

biomGIRO <- c(biomGIRO1,
              biomGIRO2,
              rnbinom(1, 0.44, mu = 2376))

biomLECY <- c(biomLECY1,
              biomLECY2,
              rnbinom(1, 0.34, mu = 164))

biomCAIN <- c(biomCAIN1,
              biomCAIN2,
              rnbinom(1, 0.34, mu = 164))
```
biomCAIN2, 
  rnbinom(1, 1.33, mu = 34068))

biomMIDO <- c(biomMIDO1, 
  biomMIDO2, 
  rnbinom(1, 0.66, mu = 4202))

biomCYLU <- c(biomCYLU1, 
  biomCYLU2, 
  rnbinom(1, 1.78, mu = 238))

biomAMNA <- c(biomAMNA1, 
  biomAMNA2, 
  rnbinom(1, 0.36, mu = 1306))

### ---------------------------------------------------------------
### * Inner loop
### #------------------------------------------------------------------
### for(i in 1:count) {
###   # CHANGE WHAT 'y' IS TO SIMULATE DIFFERENT FLOW REGIMES
###   ACROSS THE 54 Y
###   y = i # follow flow record sequence
###   # Sampling randomly from the flow record
###   # y = sample(nrow(flowdata), 1)
###   # Transition probabilities
###   # G is prob. of transition to next stage
###   # P is prob. of remaining in that stage
###   # Baseline mortality vital rate (from file object: 'vital-rates.csv')
###   # 'STmort...' is multiplied by modifier (from file object:
###   # 'modifiers-all-species.csv') based on yeartype as specified above
###   # So we have SP_highflood, SU_highflood, medflood, nonevent, drought
###   # and '..J/2/3_SUHF/SPHF/MF/NE/DR'
## Stage 1 - G
for(nm in sppnames) {
    assign(paste0('G', nm, '1'),
        get(paste0('a', nm, '1')) *
        get(paste0('den', nm, 'J')) *
        (1 /
            get(paste0('den', nm, '2'))
        ) *
        (1 -
            (flowdata$SU_highflood[y] *
                get(paste0('STMort', nm)) *
                get(paste0(nm, '_J_SUHF'))) *
        (1 -
            (flowdata$SP_highflood[y] *
                get(paste0('STMort', nm)) *
                get(paste0(nm, '_J_SPHF'))) *
        (1 -
            (flowdata$medflood[y] *
                get(paste0('STMort', nm)) *
                get(paste0(nm, '_J_MF'))) *
        (1 -
            (flowdata$nonevent[y] *
                get(paste0('STMort', nm)) *
                get(paste0(nm, '_J_NE'))) *
        (1 -
            (flowdata$drought[y] *
                get(paste0('STMort', nm)) *
                get(paste0(nm, '_J_DR'))) )
    )
}

## Stage 2 - G
for(nm in sppnames) {
    assign(paste0('G', nm, '2'),
        get(paste0('a', nm, '2')) *
        get(paste0('den', nm, '2')) *
        (1 /
            get(paste0('den', nm, '2'))
        ) *
        (1 -
            (flowdata$SU_highflood[y] *
                get(paste0('STMort', nm)) *
                get(paste0(nm, '_J_SUHF'))) *
        (1 -
            (flowdata$SP_highflood[y] *
                get(paste0('STMort', nm)) *
                get(paste0(nm, '_J_SPHF'))) *
        (1 -
            (flowdata$medflood[y] *
                get(paste0('STMort', nm)) *
                get(paste0(nm, '_J_MF'))) *
        (1 -
            (flowdata$nonevent[y] *
                get(paste0('STMort', nm)) *
                get(paste0(nm, '_J_NE'))) *
        (1 -
            (flowdata$drought[y] *
                get(paste0('STMort', nm)) *
                get(paste0(nm, '_J_DR'))) )
    )
}
get(paste0('den', nm, '3'))
)
*
(1 -
(flowdata$SU_highflood[y] *
 get(paste0('STMort', nm)) *
 get(paste0(nm, '_A_SUHF'))) *
(1 -
(flowdata$SP_highflood[y] *
 get(paste0('STMort', nm)) *
 get(paste0(nm, '_A_SPHF'))) *
(1 -
(flowdata$medflood[y] *
 get(paste0('STMort', nm)) *
 get(paste0(nm, '_A_MF'))) *
(1 -
(flowdata$nonevent[y] *
 get(paste0('STMort', nm)) *
 get(paste0(nm, '_A_NE'))) *
(1 -
(flowdata$drought[y] *
 get(paste0('STMort', nm)) *
 get(paste0(nm, '_A_DR'))) } )

## Stage 3 - P
for(nm in sppnames) {
  assign(paste0('P', nm, '3'),

    (1 -
     get(paste0('a', nm, '3'))
    ) *
(1 -
 (flowdata$SU_highflood[y] *
   get(paste0('STMort', nm)) *
   get(paste0(nm, '_A_SUHF'))) *
(1 -
 (flowdata$SP_highflood[y] *
   get(paste0('STMort', nm)) *
   get(paste0(nm, '_A_SPHF'))) *
(1 -
 (flowdata$medflood[y] *
   get(paste0('STMort', nm)) *
   get(paste0(nm, '_A_MF'))) *
(1 -
 (flowdata$nonevent[y] *
   get(paste0('STMort', nm)) *
   get(paste0(nm, '_A_NE'))) *
(1 -
 (flowdata$drought[y] *
   get(paste0('STMort', nm)) *
   get(paste0(nm, '_A_DR'))) )}
get(paste0('STMort', nm)) *
get(paste0(nm, '_A_SPHF'))

(1 -
(flowdata$medflood[y] *
get(paste0('STMort', nm)) *
get(paste0(nm, '_A_MF'))
)

(1 -
(flowdata$nonevent[y] *
get(paste0('STMort', nm)) *
get(paste0(nm, '_A_NE'))
)

(1 -
(flowdata$drought[y] *
get(paste0('STMort', nm)) *
get(paste0(nm, '_A_DR'))
)
)

}

## POTENTIAL FECUNDITY ------------------------------------------

## 1st calculate total grams occupied after year
totbiom <- ldply(sppnames, function(x)
    get(paste0('biom', x))[1] +
    get(paste0('biom', x))[2] +
    get(paste0('biom', x))[3]) %>%
  sum

## Carrying capacity (K) is limiting spawning of all species
## based on
## the total biomass occupied at the end of the previous year.
## i.e. if
## above K, no spp spawn in that year. If spawning occurs,
## they all do.
## Some slight differences in fecund b/wn spp so can't
## loop/lapply

## CAACL stage 2
FCACL2 <- ((0.5 * GSI.CACL * (1 - S0MortCAACL)) *
  checkpos((K - totbiom)/K)) *
  denCAACL1 *
(1/denCACLJ)

## CACL stage 3
FCACL3 <- ((0.5 * GSI.CACL * (1 - S0MortCACL)) *
    checkpos((K - totbiom)/K)) *
    denCACL1 *
    (1/denCACLJ)

## GIRO stage 2
FGIRO2 <- ((0.5 * GSI.GIRO * (1 - S0MortGIRO)) *
    checkpos((K - totbiom)/K)) *
    denGIRO1 *
    (1/denGIROJ)

## GIRO stage 3
FGIRO3 <- ((0.5 * GSI.GIRO * (1 - S0MortGIRO)) *
    checkpos((K - totbiom)/K)) *
    denGIRO1 *
    (1/denGIROJ)

## CAIN stage 3
FCAIN3 <- ((0.5 * GSI.CAIN * (1 - S0MortCAIN)) *
    checkpos((K - totbiom)/K)) *
    denCAIN1 *
    (1/denCAINJ)

## LECY stage 2
FLECY2 <- ((0.5 * GSI.LECY * (1 - S0MortLECY)) *
    checkpos((K - totbiom)/K)) *
    denLECY1 *
    (1/denLECYJ)

## LECY stage 3
FLECY3 <- ((0.5 * GSI.LECY * (1 - S0MortLECY)) *
    checkpos((K - totbiom)/K)) *
    denLECY1 *
    (1/denLECYJ)

## MIDO stage 3
FMIDO3 <- ((0.5 * GSI.MIDO * (1 - S0MortMIDO)) *
    checkpos((K - totbiom)/K)) *
    denMIDO1 *
    (1/denMIDOJ)
## CYLU

because they are serial spawners, they are allowed to spawn twice a season in stage 2 and 3

## CYLU stage 1

FCYLUJ <- ((0.5 * GSI.CYLU * (1 - S0MortCYLU)) *
    checkpos((K - totbiom)/K)) *
    denCYLU1 *
    (1/denCYLUJ)

## CYLU stage 2

FCYLU2 <- ((0.5 * 2 * GSI.CYLU * (1 - S0MortCYLU)) *
    checkpos((K - totbiom)/K)) *
    denCYLU1 *
    (1/denCYLUJ)

## CYLU stage 3

FCYLU3 <- ((0.5 * 2 * GSI.CYLU * (1 - S0MortCYLU)) *
    checkpos((K - totbiom)/K)) *
    denCYLU1 *
    (1/denCYLUJ)

## AMNA

FAMNA3 <- ((0.5 * GSI.AMNA * (1 - S0MortAMNA)) *
    checkpos((K - totbiom)/K)) *
    denAMNA1 *
    (1/denAMNAJ)

## K

Calculating the percentage of K occupied and adding to KCACL etc

for(nm in sppnames) {
    assign(paste0('K', nm), 100 * (get(paste0('biom', nm))[1] +
    get(paste0('biom', nm))[2] +
    get(paste0('biom', nm))[3])/K)
}

## TRANSITION MATRICES

## CACL

ACACL1 <- c(0, FCACL2, FCACL3)
ACACL2 <- c(GCACL1, 0, 0)
ACACL3 <- c(0, GCACL2, PCACL3)

## GIRO
AGIRO1 <- c(0, FGIRO2, FGIRO3)
AGIRO2 <- c(GGIRO1, 0, 0)
AGIRO3 <- c(0, GGIRO2, PGIRO3)

## LECY
ALECY1 <- c(0, FLECY2, FLECY3)
ALECY2 <- c(GLECY1, 0, 0)
ALECY3 <- c(0, GLECY2, PLECY3)

## CAIN
ACAIN1 <- c(0, 0, FCAIN3)
ACAIN2 <- c(GCAIN1, 0, 0)
ACAIN3 <- c(0, GCAIN2, PCAIN3)

## MIDO
AMIDO1 <- c(0, 0, FMIDO3)
AMIDO2 <- c(GMIDO1, 0, 0)
AMIDO3 <- c(0, GMIDO2, PMIDO3)

## CYLU
ACYLU1 <- c(FCYLU1, FCYLU2, FCYLU3)
ACYLU2 <- c(GCYLU1, 0, 0)
ACYLU3 <- c(0, GCYLU2, PCYLU3)

## AMNA
AAMNA1 <- c(0, 0, FAMNA3)
AAMNA2 <- c(GAMNA1, 0, 0)
AAMNA3 <- c(0, GAMNA2, PAMNA3)

## rbinding the vectors from above into transition matrices
## Makes ACACL, AGIRO etc.
for(nm in sppnames) {
    assign(paste0('A', nm), rbind(
        get(paste0('A', nm, '1')),
        get(paste0('A', nm, '2')),
        get(paste0('A', nm, '3'))
    ))
}
## COMPILING OUTPUTS

### Lambda values
Filling in the df with lambda values for each species and each year
Species as columns, years as rows
This applies 'lambda(ACACL)' etc and adds to correct column each 'i' value (year)
\[
\text{lambda.df}[i,] \leftarrow \text{sapply(mget(paste0('A', names(lambda.df))), lambda)}
\]

### Fecundity values
Cant loop or anything as different for diff spp
\[
\begin{align*}
\text{fec.list}$\text{CACL}[i] & \leftarrow FCACL3 + FCACL2 \\
\text{fec.list}$\text{GIRO}[i] & \leftarrow FGIRO3 + FGIRO2 \\
\text{fec.list}$\text{LECY}[i] & \leftarrow FLECY3 + FLECY2 \\
\text{fec.list}$\text{CAIN}[i] & \leftarrow FCAIN3 \\
\text{fec.list}$\text{MIDO}[i] & \leftarrow FMIDO3 \\
\text{fec.list}$\text{CYLU}[i] & \leftarrow FCYLU3 + FCYLU2 + FCYLUJ \\
\text{fec.list}$\text{AMNA}[i] & \leftarrow FAMNA3
\end{align*}
\]

### biomass values into each df/array in the list
\[
\text{for(nm in sppnames) } \{ \\
\text{output.biom.list}[[nm]][i,1:3] \leftarrow \text{get(paste0('biom', nm))}
\}
\]

### N values into each df/array in the list
\[
\text{for(nm in sppnames) } \{ \\
\text{output.N.list}[[nm]][i,1:3] \leftarrow \text{c(get(paste0('biom', nm))[1]} \\
\text{get(paste0('den', nm, 'J'))}, \\
\text{get(paste0('biom', nm))[2]} \\
\text{get(paste0('den', nm, '2'))}, \\
\text{get(paste0('biom', nm))[3]} \\
\text{get(paste0('den', nm, '3'))}
\}
\]
## Flow results

### Records flood settings of each particular projected year

(1 = yes, 0 = no)

\[
\begin{align*}
\text{flowresults}\$SPhighflood[i] & < \text{flowdata}\$SP\_highflood[y] \\
\text{flowresults}\$SUhighflood[i] & < \text{flowdata}\$SU\_highflood[y] \\
\text{flowresults}\$medflood[i] & < \text{flowdata}\$medflood[y] \\
\text{flowresults}\$nonevent[i] & < \text{flowdata}\$nonevent[y] \\
\text{flowresults}\$drought[i] & < \text{flowdata}\$drought[y]
\end{align*}
\]

### MATRIX MULTIPLICATION

---

can include rescue function for each with 0.5 chance of reach being colonized by 2 individuals

Loop essentially == 

\[
\text{biomAMNA} \left< \text{AAMNA} \times \text{biomAMNA}
\right>
\]

AAMNA is transit. matrix, biomAMNA = total biomass for each age class

for(nm in sppnames) {
  assign(paste0('biom', nm), 
        get(paste0('A', nm)) \times get(paste0('biom', nm))
  )
}

---

### End of inner loop

----------


### Mean values for each iteration run over each sequence of years

---

for(nm in sppnames) {
  replist[[nm]][,,iter] <- output.N.list[[nm]]
}

---

## Total.N

---

## Calculating Total.N for each year, and adding it to total.N data frame

with however many iterations run.

Total does not incl. juveniles.
## map is purrr version of lapply. Can pass fn using ~ and .x
instead of
## function(x) x
## Gets list output of stages 2:3 for ea spp, then cbinds them all
together,
## then calcs sum.
Total.N[,iter] <- map(output.N.list, ~ .x[,2:3]) %>%
do.call('cbind', .) %>%
apply(1, sum)

### End of mid loop

### Saving image here - pre compiling results
save.image()

### * OUTPUTS

---

### Compiling abundance and biomass outputs into single dfs

#### Biomass
## Compiling df from output.biom.list, renaming cols to stages, adding a
## replicate col and gathering into long form
ALLoutput.biom.DF <- ldply(output.biom.list, function(x) {
    as.data.frame(x) %>%
    rename(S1 = V1, S2 = V2, S3 = V3) %>%
    mutate(rep = row.names(.)) %>%
gather(stage, val, -rep)
}) %>%
rename(spp = `.id`, g = val)

## Abundance
ALLoutput.N.DF <- ldply(output.N.list, function(x) {
```r
as.data.frame(x) %>%
  rename(S1 = V1, S2 = V2, S3 = V3) %>%
  mutate(rep = row.names(.)) %>%
  gather(stage, val, -rep)
)
rename(spp = `.id`, N = val)

## Graph biomass
ggplot(ALLoutput.biom.DF, aes(as.numeric(rep), g, colour = stage)) +
  geom_point() +
  geom_path() +
  facet_grid(stage~spp, scales = "free")

## Graph abundance
ggplot(ALLoutput.N.DF, aes(as.numeric(rep), N, colour = stage)) +
  geom_point() +
  geom_path() +
  facet_grid(stage~spp, scales = "free")

## Graph all species together
ggplot(ALLoutput.biom.DF, aes(as.numeric(rep), g, colour = stage)) +
  geom_point() +
  geom_path() +
  facet_grid(~spp)

## Graph all
ggplot(ALLoutput.N.DF, aes(as.numeric(rep), N, colour = stage)) +
  geom_point() +
  geom_path() +
  facet_grid(~spp)

## Graph flows
flowresults.l <- flowresults %>%
  mutate(rep = as.numeric(row.names(.))) %>%
  gather(metric, value, -rep)

ggplot(flowresults.l, aes(rep, value)) +
  geom_point() + geom_path() +
  facet_wrap(~metric)

## Checking to see if flows used in actual model runs match those input.
## This current run uses natural flow only.
flowtest <- data.frame(cbind(flowdata$SP_highflood, flowdata$SU_highflood,
                        flowdata$SU_highflood,
```
flowdata$medflood,
flowdata$drought,
flowdata$nonevent)

flowtest

apply(flowtest, 1, sum)

flowresults
flowtest[,1]-flowresults[,1]
flowtest[,2]-flowresults[,2]
flowtest[,3]-flowresults[,3]
flowtest[,4]-flowresults[,4]
flowtest[,5]-flowresults[,5]

## Plot summary from all iterations of model run and compare to relative abundance from observed surveys

## Reading in observed field data
Verde <- read.csv("data/Rel_Abu_Verde_94-08.csv", header = T)

## renaming as observed, removing tot abund, and renaming cols
observed <- Verde %>%
  select(year = Year,
        species = SppCode,
        obs.mean.rel.abund = MeanRelAbu,
        obs.se.rel.abund = SERelAbu)
observed$year <- as.numeric(as.character(observed$year))

## turning replist into a df
repdf <- ldply(replist, function(x) {
  adply(x, c(1,2,3))
})

names(repdf) <- c('species', 'year', 'stage', 'rep', 'abund')
repdf <- filter(repdf, stage != 'S1')
repdf$year <- as.numeric(as.character(repdf$year))

totn <- adply(Total.N, c(1,2))
names(totn) <- c('year', 'rep', 'tot.abund')
totn$year <- as.numeric(as.character(totn$year))

## joining totn and repdf together
repdf <- left_join(totn, repdf)

## calculating relative abundance
repdf <- mutate(repdf, rel.abund = abund/tot.abund)

## Taking mean results to cf w/ observed data
means <- repdf %>%
  select(-tot.abund) %>%
  group_by(year, rep, species) %>% # combining stages
  summarise(abund = sum(abund),
    rel.abund = sum(rel.abund)) %>%
  ungroup() %>%
  group_by(species, year) %>%
  summarise(mean.abund = mean(abund),
    sd.abund = sd(abund),
    se.abund = sd(abund)/sqrt(iterations),
    mean.rel.abund = mean(rel.abund),
    sd.rel.abund = sd(rel.abund),
    se.rel.abund = sd(rel.abund)/sqrt(iterations)) %>%
  ungroup()
means

## Taking the end period to compare with observed data
mean_end <- filter(means, year >= 1994)

## Joining w/ observed data
mean_end <- left_join(mean_end, observed)

## Plotting model vs. observed for 1994-2017
rel.abund.trends <- ggplot(mean_end, aes(year, mean.rel.abund, colour = species, fill = species)) +
  geom_ribbon(aes(ymin = mean.rel.abund - 1.96 * se.rel.abund, ymax = mean.rel.abund + 1.96 * se.rel.abund),
    colour = 'transparent',
    alpha = .5,
    show.legend = FALSE) +
  geom_line(show.legend = FALSE) +
  facet_wrap(~species, ncol = 2) +
  theme_classic_facet() +
coord_cartesian(ylim = c(0,1)) +
ylab('Relative abundance') +
xlab('Year')
## adding observed data
rel.abund.trends +
    geom_pointrange(aes(y = obs.mean.rel.abund,
        ymin = obs.mean.rel.abund - 1.96 *
        obs.se.rel.abund,
        ymax = obs.mean.rel.abund + 1.96 *
        obs.se.rel.abund),
        size = .1,
        show.legend = FALSE)
ggsave('export/multi-spp2.pdf', width = 4, height = 6)
## * Correlation tests
-----------------------------------------------
## Create a df w/ model and observed relative abundances from 1994-2008 to test
## correlation between them
spearman.results <- mean_end %>%
    filter(year >= 1994, year <= 2008) %>%
    group_by(year) %>%
    summarise(rho = cor.test(mean.rel.abund,
        obs.mean.rel.abund,
        method = 'spearman')$estimate,
        pval = cor.test(mean.rel.abund,
        obs.mean.rel.abund,
        method = 'spearman')$p.value )
spearman.results

## Overall correlation between mean observed and modeled relative abund --
mod.obs.mean.by.spp <- mean_end %>%
    filter(year >= 1994, year <= 2008) %>%
    group_by(species) %>%
    summarise(model = mean(mean.rel.abund),
obs = \texttt{mean(obs.mean.rel.abund)}

\begin{verbatim}
mod.obs.mean.by.spp \%>%
s summarise(rho = cor.test(model,
    obs,
    method = \texttt{\'spearman\')}$estimate,

    pval = cor.test(model,
    obs,
    method = \texttt{\'spearman\')}$p.value
)
\end{verbatim}

## Species level correlations ----------------------------------------
##
## Spearmans
mean_end \%>%
 filter(year >= 1994, year <= 2008) \%>%
 select(species, year, mean.rel.abund, obs.mean.rel.abund) \%>%
 group_by(species) \%>%
 summarise(spear.rho = cor.test(mean.rel.abund,
    obs.mean.rel.abund,
    method = \texttt{\'spearman\')}$estimate,

    spear.pval = cor.test(mean.rel.abund,
    obs.mean.rel.abund,
    method = \texttt{\'spearman\')}$p.value
)
##
## Pearsons
mean_end \%>%
 filter(year >= 1994, year <= 2008) \%>%
 select(species, year, mean.rel.abund, obs.mean.rel.abund) \%>%
 group_by(species) \%>%
 summarise(pear.r = cor.test(mean.rel.abund,
    obs.mean.rel.abund,
    method = \texttt{\'pearson\')}$estimate,

    pear.pval = cor.test(mean.rel.abund,
    obs.mean.rel.abund,
    method = \texttt{\'pearson\')}$p.value
)
##
## Saving current state
save.image()

### Local Variables:
### eval: (orgstruct-mode 1)
### orgstruct-heading-prefix-regexp: "## \\
### End:

Test dataset for fish model:
https://zenodo.org/record/1309024#.XitmFhPYrVo

<table>
<thead>
<tr>
<th>Year</th>
<th>SppCode</th>
<th>MeanRelAbu</th>
<th>SERelAbu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1994</td>
<td>AMNA</td>
<td>0.00474129</td>
<td>0.0015367</td>
</tr>
<tr>
<td>1995</td>
<td>AMNA</td>
<td>0.02369851</td>
<td>0.00659485</td>
</tr>
<tr>
<td>1996</td>
<td>AMNA</td>
<td>0.01542454</td>
<td>0.00484929</td>
</tr>
<tr>
<td>1997</td>
<td>AMNA</td>
<td>0.01782863</td>
<td>0.00909657</td>
</tr>
<tr>
<td>1998</td>
<td>AMNA</td>
<td>0.04751465</td>
<td>0.01304762</td>
</tr>
<tr>
<td>1999</td>
<td>AMNA</td>
<td>0.02869334</td>
<td>0.00733422</td>
</tr>
<tr>
<td>2000</td>
<td>AMNA</td>
<td>0.01625465</td>
<td>0.00667679</td>
</tr>
<tr>
<td>2001</td>
<td>AMNA</td>
<td>0.02373947</td>
<td>0.00765028</td>
</tr>
<tr>
<td>2002</td>
<td>AMNA</td>
<td>0.02755906</td>
<td>0.0194872</td>
</tr>
<tr>
<td>2003</td>
<td>AMNA</td>
<td>0.01556787</td>
<td>0.00542145</td>
</tr>
<tr>
<td>2004</td>
<td>AMNA</td>
<td>0.01701088</td>
<td>0.00582928</td>
</tr>
<tr>
<td>2005</td>
<td>AMNA</td>
<td>0.06875478</td>
<td>0.04801822</td>
</tr>
<tr>
<td>2006</td>
<td>AMNA</td>
<td>0.01820826</td>
<td>0.00520848</td>
</tr>
<tr>
<td>2007</td>
<td>AMNA</td>
<td>0.03008951</td>
<td>0.00462392</td>
</tr>
<tr>
<td>2008</td>
<td>AMNA</td>
<td>0.01882535</td>
<td>0.00562542</td>
</tr>
<tr>
<td>1994</td>
<td>CACL</td>
<td>0.36355263</td>
<td>0.02473104</td>
</tr>
<tr>
<td>1995</td>
<td>CACL</td>
<td>0.31718884</td>
<td>0.04181266</td>
</tr>
<tr>
<td>1996</td>
<td>CACL</td>
<td>0.31718884</td>
<td>0.04181266</td>
</tr>
<tr>
<td>1997</td>
<td>CACL</td>
<td>0.20041592</td>
<td>0.05741031</td>
</tr>
<tr>
<td>1998</td>
<td>CACL</td>
<td>0.15306896</td>
<td>0.03719239</td>
</tr>
<tr>
<td>1999</td>
<td>CACL</td>
<td>0.19768335</td>
<td>0.034577</td>
</tr>
<tr>
<td>2000</td>
<td>CACL</td>
<td>0.05151472</td>
<td>0.01629347</td>
</tr>
<tr>
<td>2001</td>
<td>CACL</td>
<td>0.17419184</td>
<td>0.05427227</td>
</tr>
<tr>
<td>2002</td>
<td>CACL</td>
<td>0.09934383</td>
<td>0.04667647</td>
</tr>
<tr>
<td>2003</td>
<td>CACL</td>
<td>0.09034051</td>
<td>0.04579828</td>
</tr>
<tr>
<td>2004</td>
<td>CACL</td>
<td>0.06695886</td>
<td>0.02289526</td>
</tr>
<tr>
<td>2005</td>
<td>CACL</td>
<td>0.1527301</td>
<td>0.06418209</td>
</tr>
<tr>
<td>2006</td>
<td>CACL</td>
<td>0.34151287</td>
<td>0.04582022</td>
</tr>
<tr>
<td>2007</td>
<td>CACL</td>
<td>0.32437406</td>
<td>0.10218552</td>
</tr>
<tr>
<td>2008</td>
<td>CACL</td>
<td>0.21786307</td>
<td>0.07327176</td>
</tr>
<tr>
<td>1994</td>
<td>CAIN</td>
<td>0.33618776</td>
<td>0.05901669</td>
</tr>
<tr>
<td>1995</td>
<td>CAIN</td>
<td>0.31572465</td>
<td>0.04577041</td>
</tr>
<tr>
<td>Year</td>
<td>Entity</td>
<td>Value 1</td>
<td>Value 2</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>1996</td>
<td>CAIN</td>
<td>0.33950901</td>
<td>0.03753834</td>
</tr>
<tr>
<td>1997</td>
<td>CAIN</td>
<td>0.14186852</td>
<td>0.04088214</td>
</tr>
<tr>
<td>1998</td>
<td>CAIN</td>
<td>0.1539453</td>
<td>0.03651483</td>
</tr>
<tr>
<td>1999</td>
<td>CAIN</td>
<td>0.13942865</td>
<td>0.0397454</td>
</tr>
<tr>
<td>2000</td>
<td>CAIN</td>
<td>0.11702555</td>
<td>0.05773223</td>
</tr>
<tr>
<td>2001</td>
<td>CAIN</td>
<td>0.06639228</td>
<td>0.02344621</td>
</tr>
<tr>
<td>2002</td>
<td>CAIN</td>
<td>0.07362205</td>
<td>0.01865203</td>
</tr>
<tr>
<td>2003</td>
<td>CAIN</td>
<td>0.05831369</td>
<td>0.02539519</td>
</tr>
<tr>
<td>2004</td>
<td>CAIN</td>
<td>0.06853656</td>
<td>0.0206245</td>
</tr>
<tr>
<td>2005</td>
<td>CAIN</td>
<td>0.08974169</td>
<td>0.03180243</td>
</tr>
<tr>
<td>2006</td>
<td>CAIN</td>
<td>0.22810135</td>
<td>0.03261757</td>
</tr>
<tr>
<td>2007</td>
<td>CAIN</td>
<td>0.246846</td>
<td>0.06974082</td>
</tr>
<tr>
<td>2008</td>
<td>CAIN</td>
<td>0.30335155</td>
<td>0.08384131</td>
</tr>
<tr>
<td>1994</td>
<td>CYLU</td>
<td>0.19513838</td>
<td>0.05095858</td>
</tr>
<tr>
<td>1995</td>
<td>CYLU</td>
<td>0.10965148</td>
<td>0.04324903</td>
</tr>
<tr>
<td>1996</td>
<td>CYLU</td>
<td>0.11786913</td>
<td>0.02877738</td>
</tr>
<tr>
<td>1997</td>
<td>CYLU</td>
<td>0.52653933</td>
<td>0.13207607</td>
</tr>
<tr>
<td>1998</td>
<td>CYLU</td>
<td>0.46119419</td>
<td>0.10330325</td>
</tr>
<tr>
<td>1999</td>
<td>CYLU</td>
<td>0.43415439</td>
<td>0.09697001</td>
</tr>
<tr>
<td>2000</td>
<td>CYLU</td>
<td>0.72122078</td>
<td>0.0786306</td>
</tr>
<tr>
<td>2001</td>
<td>CYLU</td>
<td>0.47704808</td>
<td>0.13264191</td>
</tr>
<tr>
<td>2002</td>
<td>CYLU</td>
<td>0.26692913</td>
<td>0.09409531</td>
</tr>
<tr>
<td>2003</td>
<td>CYLU</td>
<td>0.56485803</td>
<td>0.03661946</td>
</tr>
<tr>
<td>2004</td>
<td>CYLU</td>
<td>0.71210731</td>
<td>0.04081048</td>
</tr>
<tr>
<td>2005</td>
<td>CYLU</td>
<td>0.57879833</td>
<td>0.11942473</td>
</tr>
<tr>
<td>2006</td>
<td>CYLU</td>
<td>0.28684651</td>
<td>0.07329326</td>
</tr>
<tr>
<td>2007</td>
<td>CYLU</td>
<td>0.28979869</td>
<td>0.10017037</td>
</tr>
<tr>
<td>2008</td>
<td>CYLU</td>
<td>0.22581426</td>
<td>0.08361998</td>
</tr>
<tr>
<td>1994</td>
<td>GIRO</td>
<td>0.09705336</td>
<td>0.02366</td>
</tr>
<tr>
<td>1995</td>
<td>GIRO</td>
<td>0.19873357</td>
<td>0.04229313</td>
</tr>
<tr>
<td>1996</td>
<td>GIRO</td>
<td>0.17723518</td>
<td>0.03449633</td>
</tr>
<tr>
<td>1997</td>
<td>GIRO</td>
<td>0.0375248</td>
<td>0.01620933</td>
</tr>
<tr>
<td>1998</td>
<td>GIRO</td>
<td>0.05669829</td>
<td>0.01805462</td>
</tr>
<tr>
<td>1999</td>
<td>GIRO</td>
<td>0.03922614</td>
<td>0.0147227</td>
</tr>
<tr>
<td>2000</td>
<td>GIRO</td>
<td>0.01400304</td>
<td>0.00665164</td>
</tr>
<tr>
<td>2001</td>
<td>GIRO</td>
<td>0.04192696</td>
<td>0.02243223</td>
</tr>
<tr>
<td>2002</td>
<td>GIRO</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>2003</td>
<td>GIRO</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>2004</td>
<td>GIRO</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>2005</td>
<td>GIRO</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>Year</td>
<td>Code</td>
<td>Value1</td>
<td>Value2</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>1994</td>
<td>GIRO</td>
<td>0.06814602</td>
<td>0.02737813</td>
</tr>
<tr>
<td>1995</td>
<td>GIRO</td>
<td>0.03393931</td>
<td>0.02908034</td>
</tr>
<tr>
<td>1996</td>
<td>GIRO</td>
<td>0.02661457</td>
<td>0.01066999</td>
</tr>
<tr>
<td>1997</td>
<td>GIRO</td>
<td>0.0277769</td>
<td>0.01576466</td>
</tr>
<tr>
<td>1998</td>
<td>GIRO</td>
<td>0.03324574</td>
<td>0.01702813</td>
</tr>
<tr>
<td>1999</td>
<td>GIRO</td>
<td>0.02936869</td>
<td>0.01970524</td>
</tr>
<tr>
<td>2000</td>
<td>GIRO</td>
<td>0.04948392</td>
<td>0.02375537</td>
</tr>
<tr>
<td>2001</td>
<td>GIRO</td>
<td>0.09059527</td>
<td>0.05744906</td>
</tr>
<tr>
<td>2002</td>
<td>GIRO</td>
<td>0.11463255</td>
<td>0.06927234</td>
</tr>
<tr>
<td>2003</td>
<td>GIRO</td>
<td>0.08863909</td>
<td>0.03731152</td>
</tr>
<tr>
<td>2004</td>
<td>GIRO</td>
<td>0.02951</td>
<td>0.01300644</td>
</tr>
<tr>
<td>2005</td>
<td>GIRO</td>
<td>0.05923339</td>
<td>0.01494385</td>
</tr>
<tr>
<td>2006</td>
<td>GIRO</td>
<td>0.00830282</td>
<td>0.00219272</td>
</tr>
<tr>
<td>2007</td>
<td>GIRO</td>
<td>0.03035992</td>
<td>0.01601996</td>
</tr>
<tr>
<td>2008</td>
<td>GIRO</td>
<td>0.0885592</td>
<td>0.03250162</td>
</tr>
<tr>
<td>1994</td>
<td>LECY</td>
<td>0.00307949</td>
<td>0.00094557</td>
</tr>
<tr>
<td>1995</td>
<td>LECY</td>
<td>0.00838838</td>
<td>0.00348347</td>
</tr>
<tr>
<td>1996</td>
<td>LECY</td>
<td>0.03591548</td>
<td>0.01434669</td>
</tr>
<tr>
<td>1997</td>
<td>LECY</td>
<td>0.04804591</td>
<td>0.03089934</td>
</tr>
<tr>
<td>1998</td>
<td>LECY</td>
<td>0.09433287</td>
<td>0.02663666</td>
</tr>
<tr>
<td>1999</td>
<td>LECY</td>
<td>0.13144544</td>
<td>0.0305531</td>
</tr>
<tr>
<td>2000</td>
<td>LECY</td>
<td>0.03049735</td>
<td>0.00944073</td>
</tr>
<tr>
<td>2001</td>
<td>LECY</td>
<td>0.1261061</td>
<td>0.0469021</td>
</tr>
<tr>
<td>2002</td>
<td>LECY</td>
<td>0.41791339</td>
<td>0.02268866</td>
</tr>
<tr>
<td>2003</td>
<td>LECY</td>
<td>0.17919201</td>
<td>0.02800196</td>
</tr>
<tr>
<td>2004</td>
<td>LECY</td>
<td>0.10233099</td>
<td>0.03465866</td>
</tr>
<tr>
<td>2005</td>
<td>LECY</td>
<td>0.05074172</td>
<td>0.03476232</td>
</tr>
<tr>
<td>2006</td>
<td>LECY</td>
<td>0.05260674</td>
<td>0.01876522</td>
</tr>
<tr>
<td>2007</td>
<td>LECY</td>
<td>0.04459252</td>
<td>0.01427386</td>
</tr>
<tr>
<td>2008</td>
<td>LECY</td>
<td>0.13107796</td>
<td>0.03760383</td>
</tr>
</tbody>
</table>
Invertebrate population model

Primary publication describing the methodology:
https://doi.org/10.1111/ele.12866

Mathematical proof of the time-varying logistic model:

Logistic equation solution that allows time-varying growth rate and carrying capacity

Consider the logistic equation with time-varying maximal per capita growth rate $r(t)$, and time-varying carrying capacity $K(t)$:

$$
\frac{dN}{dt}(t) = r(t)N(t) \left(1 - \frac{N(t)}{K(t)}\right), \quad N(t_0) = N_0
$$

(4)

Here $t_0$ is a given initial time, and $N_0 > 0$ is the initial population size. We assume that $r(t)$ and $K(t)$ are positive, bounded, piecewise continuous functions, defined on the interval $[t_0, +\infty)$. (A function $f(t)$, defined on the interval $[t_0, +\infty)$ is said to be piecewise continuous, if for each $t > t_0$, the restriction of $f(t)$, to the interval $[t_0, t]$, has at most finitely many points of discontinuity, and with finite right and left limits). We also assume that $K(t)$ is bounded below by $K_{\text{min}} > 0$ on the interval $[t_0, +\infty)$. These are fairly general mathematical assumptions that are satisfied in most biological contexts, and they are satisfied in all the scenarios investigated in this paper.

We claim that the solution of Eq. (4) is given by:

$$
N(t) = \frac{\left(e^{\int_{t_0}^{t} r(\tau) \, d\tau}\right)N_0}{\left(\int_{t_0}^{t} \frac{r(\tau)}{K(\tau)} \, d\tau \int_{t_0}^{\tau} r(\xi) \, d\xi\right)N_0 + 1}
$$

(5)
Notice that the assumptions made on \( r(t) \) and \( K(t) \) ensure that all the integrals appearing in Eq. (5) are well-defined. The variables \( \tau \) and \( s \) are dummy variables for integration.

To prove the validity of Eq. (5), we first introduce a new variable \( n(t) \), which is a particular scaled version of the original population size \( N(t) \):

\[
n(t) = N(t) e^{-\int_{t_0}^{t} r(\tau) d\tau}
\]  

(6)

Taking derivatives with respect to time, using the product rule for differentiation, using Eq. (4) for \( dN/dt \), and expressing the resulting equation in terms of \( n(t) \) rather than \( N(t) \) by using Eq. (6), shows that \( n(t) \) must satisfy the following differential equation:

\[
\frac{dn}{dt}(t) = -\frac{r(t)}{K(t)} e^{\int_{t_0}^{t} r(\tau) d\tau} n^2(t), \quad n(t_0) = N_0
\]  

(7)

The key point is that the latter equation is a separable differential equation, which can be solved by a standard solution method:

\[
n(t) = \frac{N_0}{\left(\int_{t_0}^{t} \frac{r(\tau)}{K(\tau)} e^{\int_{t_0}^{\tau} r(s) ds} d\tau\right)N_0 + 1}
\]  

(8)

Eq. (6) implies that \( N(t) = n(t) e^{\int_{t_0}^{t} r(\tau) d\tau} \), and substituting the above expression for \( n(t) \), shows that \( N(t) \) is indeed given by Eq. (5), as claimed.

**Methodology for obtaining vital rates for the invertebrate model:**


Estimating intrinsic rates of population increase.

Intrinsic rates of population increase (\( r \)) for the three target taxa were tabulated from age-specific fecundity and survival rates from published values using the method of Birch (1948). \( r \) was defined as the number of offspring produced by one individual per day, thus measuring the per capita rate of increase over a short time (Gotelli 1998). The parameter values used for this
approach, and the resulting estimated intrinsic rates of population increase, are given in Table B1. When information could not be found on the exact taxon of interest, information from a closely related taxon was used. When a range of values were reported, the average was used in the estimation.

*Example Calculation of Intrinsic Rate of Increase*

In order to calculate the estimated intrinsic rate of population increase, total average lifespan, number of eggs laid, and survivorship of life stages or overall survivorship were necessary (Table B1). Calculations began at the egg stage, and number of individuals were successively reduced by survivorship values until \( r \) was estimated. For example, for *Fallceon quilleri*, calculations began with eggs laid per individual (1850). The value of survivorship of eggs used was 80%, so 1850 was reduced to 1480. This was assumed to be the average number of offspring that survive to larval stage per individual. The value of survivorship of larvae used was 8.9%, so 1480 was reduced to 131.72. This was assumed to be the average number of offspring that survive to the adult stage per individual. The value of survivorship of pre-reproductive adults used was 5%, so 131.72 was reduced to 6.57. This value was divided by the mean duration of lifecycle, 28.5 days, to arrive at an estimated intrinsic rate of population increase of 0.23.

<table>
<thead>
<tr>
<th>Mayfly (<em>Fallceon quilleri</em>)</th>
<th>Dragonfly (<em>Progomphus borealis</em>)</th>
<th>Ostracod (Ostracoda)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Egg stage duration (d)</strong></td>
<td>3-34 (18.5)</td>
<td>13 to 56 (34.5)</td>
</tr>
<tr>
<td><strong>Egg stage survivorship (%)</strong></td>
<td>70-90 (80)</td>
<td>77.4</td>
</tr>
</tbody>
</table>

*Table B1.* Parameter values used for life-table calculations and resulting estimated intrinsic rates of population increase.
<table>
<thead>
<tr>
<th></th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Larval stage duration (d)</td>
<td>6-12 (9)</td>
<td>30-180 (105)</td>
<td>21-365 (193)</td>
</tr>
<tr>
<td>Larval survivorship (%)</td>
<td>8.9</td>
<td>2.53</td>
<td>-</td>
</tr>
<tr>
<td>Adult stage duration (d)</td>
<td>1</td>
<td>30</td>
<td>28-180 (104)</td>
</tr>
<tr>
<td>Adult survivorship (pre-reproductive) (%)</td>
<td>1.2-8.8 (5)</td>
<td>78.2</td>
<td>-</td>
</tr>
<tr>
<td>Overall survivorship (%)</td>
<td>-</td>
<td>-</td>
<td>40-80 (60)</td>
</tr>
<tr>
<td>Mean life cycle duration (d)</td>
<td>28.5</td>
<td>181.5</td>
<td>344.5</td>
</tr>
<tr>
<td>Eggs laid per individual</td>
<td>1200-2500 (1850)</td>
<td>1000</td>
<td>8-180 (94)</td>
</tr>
<tr>
<td>Estimated intrinsic rate of population increase (r)</td>
<td>0.23</td>
<td>0.08</td>
<td>0.16</td>
</tr>
</tbody>
</table>


**NOTE:** Mean values used in the analyses shown in parentheses.
**Literature Cited Only in Appendix B**


**R code for implementing the invertebrate model:**

[https://doi.org/10.6084/m9.figshare.5476993](https://doi.org/10.6084/m9.figshare.5476993)

```r
#`
```
## Code to run time-varying logistic population growth model for 3 aquatic invertebrates
## Date: October 2017
## Authors: Jonathan Tonkin, David Lytle, Laura McMullen, Patrick DeLeenheer
## Emails: jdtonkin@gmail.com, lytleda@oregonstate.edu, laurabethmcm@gmail.com, deleenhp@science.oregonstate.edu
## The following uses the logistic model to create flood-population response surfaces for three taxa with contrasting life histories after a single-flood event (Figure 1 in the paper).
#
=================================================================
===========================
## Setup---------------------------------------------------------
-------------------------------
## Required libraries
library(tidyverse)
library(ggplot2)
library(lattice)
library(RColorBrewer)

## Plot setup
clrs <- colorRampPalette(brewer.pal(9, "YlOrRd"))
trellis.par.set("axis.line", list(col = NA, lty = 1, lwd = 1))
theme.novpadding <- list(
    layout.heights = list(top.padding = 0, bottom.padding = 0),
    layout.widths = list(left.padding = 0, right.padding = 0)
)

## General settings--------------------------------------------------
--------------------------
## Minimum threshold of what is considered a flood
Qmin = 5

## Half saturation constant
a = 100

## Maximum flood size to run model to
Qmax = 1000

## Time to run model out to - days
t = 200

## Time zero. Can also set this in the MainLogisticSolution function instead
t0 = 0

## Main function ------------
## Function to calculate N at time t in relation to flood intensity
MainLogisticSolution <- function(r, t0, t, Q) {

    n0 = initialpopsize(Q)

    intfunc <- function(y) {
        (r / kfunc(y, Q)) * (exp(r * (y - t0)))
    }

    (exp(r * (t - t0)) * n0) / (integrate(intfunc, lower = t0, upper = t)$value * n0 + 1)
}

## Species settings -----------------------------------------------
-------------------------------
## Note that species-specific values keep the same notation with
different values further down
## e.g. g, h, r, Kd, and Kb. So it's important to run through in sequence

## Progomphus -----------------------------------------------
-------------------------------

## Values

## Rate that K returns to pre-disturbance level
g = 0.01

## Strength of disturbance-mortality relationship
h = 0.01
## Intrinsic rate of population increase
\[ r = 0.08 \]

## Kd is the carrying capacity limit following strong disturbance
\[ K_d = 100 \]

## Kb is the carrying capacity baseline when disturbances are absent
\[ K_b = 40 \]

## Functions
### Some of these differ between species depending on their relationship to disturbance

### Function to calculate the initial population size "N0" after a disturbance
\[
\text{initialpopulationsize} = \text{function}(x) \; K_b \times \exp(-h \times x)
\]

### Function to calculate the disturbance magnitude-K relationship. Sets to 0 if below the Qmin
\[
Q\text{Function} <- \text{function}(x) \begin{cases} 
\text{ifelse}(x < Q_{\text{min}}, 0, (x - Q_{\text{min}})/(a + (x - Q_{\text{min}}))) \\
\end{cases}
\]

### Function to determine K0. Carrying capacity immediately following the disturbance
\[
K_0\text{func} <- \text{function}(x) \{K_b + (K_d - K_b) \times Q\text{Function}(x)\}
\]

### Function to calc. K as a function of time post-disturbance at a particular disturbance intensity
\[
k\text{func} <- \text{function}(\tau, Q)\{ 
\text{Kb} + (K_0\text{func}(Q) - K_b) \times \exp(-g \times \tau)
\}
\]

### Checking the relationship between Q (disturbance intensity) and K0 (K post-disturbance)
\[
dfK <- \text{data.frame}(Q = \text{seq}(0, Q_{\text{max}}))
dfK <- dfK %>%
\text{mutate(K0 = K0func(Q))}
ggplot(dfK, aes(Q, K0)) + \text{geom_line()}
\]

### Creating a df for storing and examining K as a function of Q and t
\[
KQT <- \text{data.frame}(Q = \text{numeric}((Q_{\text{max}} + 1) \times (t + 1)))
\]

### Filling in Q
KQT$Q <- rep(seq(0, Qmax), each = t + 1)

## Calculating K0
KQT$K0 <- K0func(KQT$Q)

## Filling in t
KQT$t <- rep(seq(0, t), Qmax + 1)

## Calculating K
KQT$K <- Kb + (KQT$K0 - Kb) * exp(-g * KQT$t)

## Plotting K
wireframe(K ~ t + Q, data = KQT,
            aspect = c(1, .4),
            drape = TRUE,
            shade = FALSE,
            colorkey = FALSE,
            col = alpha('#ffeda0', 0.08),
            scales = list(arrows = FALSE, col = 'black'),
            screen = list(z = -40, x = -70),
            par.settings = theme.novpadding,
            col.regions = clrs(1000),
            main = 'Progomphus K')

## Creating a df to fill in with results
flowdf <- data.frame(Q = numeric((Qmax + 1) * (t + 1)))

## Filling in Q
flowdf$Q <- rep(seq(0, Qmax), each = t + 1)

## Filling in t
flowdf$t <- rep(seq(0, t), Qmax + 1)

## Calculating N at t
flowdf$Nt <- apply(flowdf, 1, function(x) MainLogisticSolution(r = r,
                                                               t0 = t0, t = x[2], Q = x[1]))

## Plotting Nt as a function of Q and t
wireframe(Nt ~ Q + t, data = flowdf,
           ylab = 'Time (days)',
           aspect = c(1, .4),
           drape = TRUE,
           shade = FALSE,
           colorkey = FALSE,
col = alpha('#ffeda0', 0.08),
scales = list(arrows = FALSE, col = 'black'),
screen = list(z = -40, x = -70),
par.settings = theme.novpadding,
col.regions = clrs(1000),
main = 'Progomphus Nt')

### Fallceon

---

## Note that Fallceon has the same model as Progomphus but different rates

## Values

## Rate that K returns to pre-disturbance level
\( g = 0.01 \)

## Strength of disturbance-mortality relationship
\( h = 0.02 \)

## Intrinsic rate of population increase
\( r = 0.23 \)

## Kd is the carrying capacity limit following strong disturbance
\( K_d = 100 \)

## Kb is the carrying capacity baseline when disturbances are absent
\( K_b = 40 \)

## Functions

### Same as Progomphus

### Function to calculate the initial population size "N0" after a disturbance
\[ \text{initialpopsize} = \text{function}(x) \ K_b \times \exp(-h \times x) \]

### Function to calculate the disturbance magnitude-K relationship
## Sets to 0 if below the Qmin
\[
\text{QFunction} <- \text{function}(x)
\quad \text{ifelse}(x < \text{Qmin}, 0, (x - \text{Qmin})/(a + (x - \text{Qmin})))
\]

### Function to determine K0. Carrying capacity immediately following the disturbance
\[ \text{K0func} <- \text{function}(x) \{ \text{Kb} + (\text{Kd} - \text{Kb}) \times \text{QFunction}(x) \} \]
## Fn to calculate K as a function of time post-disturbance at a particular disturbance intensity

```r
kfunc <- function(tau, Q){
  Kb + (K0func(Q) - Kb) * exp(-g * tau)
}
```

## Checking the relationship between Q (disturbance intensity) and K0 (K post-disturbance)

```r
dfK <- data.frame(Q = seq(0, Qmax))
dfK <- dfK %>%
  mutate(K0 = K0func(Q))
ggplot(dfK, aes(Q, K0)) + geom_line()
```

## Creating a df for storing and examining K as a function of Q and t

```r
KQT <- data.frame(Q = numeric((Qmax + 1) * (t + 1)))
```

## Filling in Q

```r
KQT$Q <- rep(seq(0, Qmax), each = t + 1)
```

## Calculating K0

```r
KQT$K0 <- K0func(KQT$Q)
```

## Filling in t

```r
KQT$t <- rep(seq(0, t), Qmax + 1)
```

## Calculating K

```r
KQT$K <- Kb + (KQT$K0 - Kb) * exp(-g * KQT$t)
```

## Plotting K

```r
wireframe(K ~ t + Q, data = KQT,
  aspect = c(1, .4),
  drape = TRUE,
  shade = FALSE,
  colorkey = FALSE,
  col = alpha('#ffeda0', 0.08),
  scales = list(arrows = FALSE, col = 'black'),
  screen = list(z = -40, x = -70),
  par.settings = theme.novpadding,
  col.regions = clrs(1000),
  main = 'Fallceon K')
```

## Creating a df to fill in with results

```r
flowdf <- data.frame(Q = numeric((Qmax + 1) * (t + 1)))
```
## Filling in Q
flowdf$Q <- rep(seq(0, Qmax), each = t + 1)

## Filling in t
flowdf$t <- rep(seq(0, t), Qmax + 1)

## Calculating N at t
flowdf$Nt <- apply(flowdf, 1, function(x) MainLogisticSolution(r = r, 
                           t0 = t0, t = x[2], Q = x[1]))

## Plotting Nt as a function of Q and t
wireframe(Nt ~ Q + t, data = flowdf,
          ylab = 'Time\n       (days)',
          aspect = c(1, .4),
          drape = TRUE,
          shade = FALSE,
          colorkey = FALSE,
          col = alpha('#ffeda0', 0.08),
          scales = list(arrows = FALSE, col = 'black'),
          screen = list(z = -40, x = -70),
          par.settings = theme.novpadding,
          col.regions = clrs(1000),
          main = 'Fallceon Nt')

### Ostracod
---------------------------------------------------------
## Note that Ostracod has a different model to the previous two

## Values

## Rate that K returns to pre-disturbance level
g = 0.01

## Strength of disturbance-mortality relationship
h = 0.05

## Intrinsic rate of population increase
r = .16

## Kd is the carrying capacity limit following strong disturbance
## Note difference to other two spp
Kd = 40
## Kb is the carrying capacity baseline when disturbances are absent

\[ Kb = 100 \]

## Functions ---------------------
## Different from two previous spp

## Function to determine K0. Carrying capacity immediately following the disturbance

\[
K0func \leftarrow \text{function}(x) \{ Kb - (Kb - Kd) \ast QFunction(x) \}
\]

## Fn to calculate K as a function of time post-disturbance at a particular disturbance intensity

\[
kfunc \leftarrow \text{function}(\tau, Q)\{
    Kb - (Kb - K0func(Q)) \ast \exp(-g \ast \tau)
\}
\]

## Checking the relationship between Q (disturbance intensity) and K0 (K post-disturbance)

\[
dfK \leftarrow \text{data.frame}(Q = \text{seq}(0, Qmax))
\]

\[
dfK \leftarrow \text{dfK} \%\% \text{mutate}(K0 = K0func(Q))
\]

\[
\text{ggplot}(dfK, \text{aes}(Q, K0)) + \text{geom_line()}
\]

## Creating a df for storing and examining K as a function of Q and t

\[
KQT \leftarrow \text{data.frame}(Q = \text{numeric}((Qmax + 1) \ast (t + 1)))
\]

## Filling in Q

\[
KQT$Q \leftarrow \text{rep}(\text{seq}(0, Qmax), \text{each} = t + 1)
\]

## Calculating K0

\[
KQT$K0 \leftarrow K0func(KQT$Q)
\]

## Filling in t

\[
KQT$t \leftarrow \text{rep}(\text{seq}(0, t), Qmax + 1)
\]

## Calculating K

\[
KQT$K \leftarrow Kb + (KQT$K0 - Kb) \ast \exp(-g \ast KQT$t)
\]

## Plotting K

\[
\text{wireframe}(K \sim t + Q, \text{data} = KQT,
\text{aspect} = c(1, .4),
\text{drape} = \text{TRUE},
\text{shade} = \text{FALSE},
\text{colorkey} = \text{FALSE})
\]
col = alpha('#ffeda0', 0.08),
scales = list(arrows = FALSE, col = 'black'),
screen = list(z = -40, x = -70),
par.settings = theme.novpadding,
col.regions = clrs(1000),
main = 'Ostracod K')

## Creating a df to fill in with results
flowdf <- data.frame(Q = numeric((Qmax + 1) * (t + 1)))

## Filling in Q
flowdf$Q <- rep(seq(0, Qmax), each = t + 1)

## Filling in t
flowdf$t <- rep(seq(0, t), Qmax + 1)

## Calculating N at t
flowdf$Nt <- apply(flowdf, 1, function(x) MainLogisticSolution(r = r,
t0 = t0, t = x[2], Q = x[1]))

## Plotting Nt as a function of Q and t
wireframe(Nt ~ Q + t, data = flowdf,
ylab = 'Time (days)',
aspect = c(1, .4),
drape = TRUE,
shade = FALSE,
colorkey = FALSE,
col = alpha('#ffeda0', 0.08),
scales = list(arrows = FALSE, col = 'black'),
screen = list(z = -40, x = -70),
par.settings = theme.novpadding,
col.regions = clrs(1000),
main = 'Ostracod Nt')

## end