aae.pop: Flexible population dynamics simulations in R

Jian D. L. Yen

Contact: jian.yen@deeca.vic.gov.au

Arthur Rylah Institute for Environmental Research, Department of Energy, Environment, and Climate Action, Victoria, Australia

Abstract

1. Matrix population models (MPMs) are commonly used to address fundamental ecological questions and guide real-world management decisions. Several open-source software packages support the fundamental analysis of MPMs, but few options exist to simulate from models that include realistic ecological complexity, such as density dependence, covariate effects, and interspecific interactions.
2. The **aae.pop** R package is a plug-and-play tool to simulate from MPMs. Originally developed for applications in aquatic ecology ("aae"), the package enables simulations of population dynamics incorporating a range of processes, such as external environmental forcing, density dependence, interspecific interactions, and harvest or release of individuals.
3. **aae.pop** provides a general framework for simulating from MPMs. This framework is described here, and illustrated with two case studies on freshwater fish populations in south-eastern Australia.
4. Models developed in **aae.pop** are easily shared, fully reproducible, and updateable. These features mean that **aae.pop** can support transparent and defensible applications of MPMs. Given its explicit focus on applications, **aae.pop** complements existing R packages for the analysis of MPMs.

Introduction

Matrix population models (MPMs) are commonly used in both theoretical and applied ecology (Caswell 2001). Matrix population models represent an ecological population as a vector of abundances in different classes (e.g., different life stages). A matrix of population vital rates governs transitions among different classes, which allows for a discrete-time representation of population dynamics as $n_{t+1} = A\, n_t$, where A is the matrix of vital rates and n_t is the vector of abundances at time t. This mathematical framework can accommodate complex ecological models, especially when the matrix A is a function ($A(x, n_t)$) of covariates x and the vector n_t (Caswell 2001).

Many studies of MPMs focus on fundamental aspects of ecology, such as life history variation or the role of perturbations in determining population dynamics (e.g., Ezard *et al.* 2010, Coste & Pavard 2020). These studies contrast with the highly applied models developed for population viability analysis (PVA), which use simulations to compare population dynamics under different scenarios (Boyce 1992). This distinction extends to software for MPMs, with many existing software packages focusing on fundamental characteristics of MPMs (e.g., the R packages popbio [Stubben & Milligan 2007], popdemo [Stott *et al.* 2012], and mpmsim [Jones 2025]), with less support for simulating population trajectories. Perhaps the most widely recognised packages for applied models are RAMAS Metapop and RAMAS GIS (Akçakaya *et al.* 1995), both of which have a license fee and are not open source. The R package steps (Visintin *et al.* 2020) provides an open-source simulation tool for spatially explicit MPMs.

Missing from existing software packages is a flexible approach to simulate from realistically complex MPMs. The **aae.pop** R package aims to address this gap. Originally developed for applications in the highly dynamic systems encountered in applied aquatic ecology ("aae"), the methods in **aae.pop** extend beyond these systems to encompass a wide range of aspatial and spatially implicit models, including metapopulations and interacting species. **aae.pop** has a deliberate focus on applications, and a key feature of the package is the flexible specification of ecological processes, such as environmental covariates, harvest or release of individuals, and multiple sources of stochasticity.

General examples

The central functions in **aae.pop** are `dynamics` and `simulate`. The `dynamics` function wraps a population matrix and associated processes into a single object. The `simulate` function takes this object and generates population projections. **aae.pop** includes efficient methods and helper functions for simulations, but users can specify their own functions for any part of the simulation process.

At a minimum, a `dynamics` object needs to contain a matrix of vital rates. However, **aae.pop** allows additional processes to be added to the `dynamics` object in a plug-and-play fashion. **aae.pop** includes options for deterministic and stochastic environmental variation, density dependence, demographic stochasticity, and additions or removals of individuals from a population. **aae.pop** additionally supports metapopulations and multispecies population dynamics.

The following sections provide a brief overview of **aae.pop**, with detailed examples available in the package vignettes. The broad framework for an **aae.pop** simulation is illustrated in Figure 1.

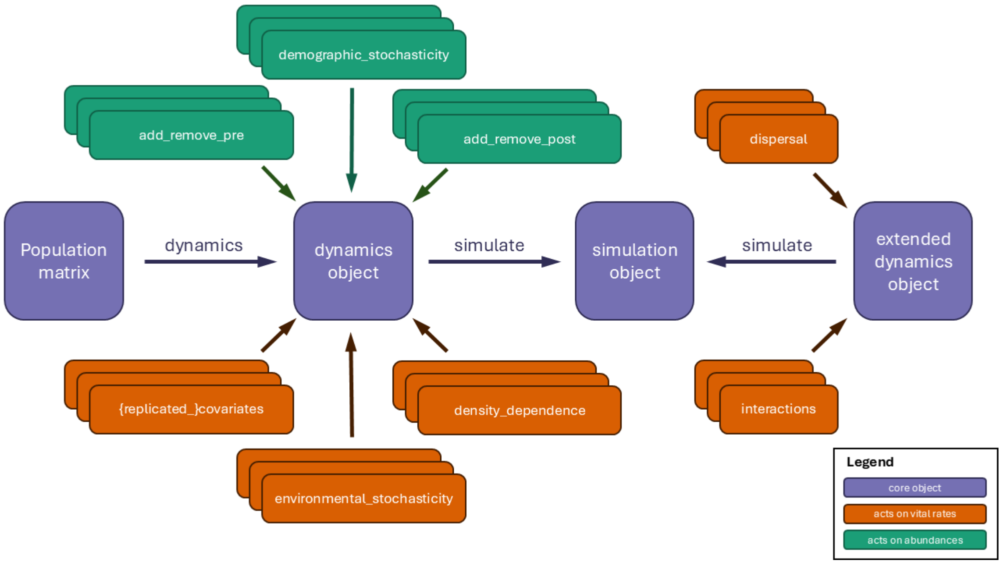

Figure 1: The conceptual framework of ***aae.pop****. A* `dynamics` *object contains the matrix of vital rates and any other processes. This object is used to* `simulate` *population trajectories. Extended* `dynamics` *objects can include dispersal (for metapopulations) and interspecific interactions (for multispecies models).*

Working with matrices in aae.pop

aae.pop is designed for generic MPMs and makes minimal assumptions about model structure or purpose. However, the package includes helper functions for common model structures, built on the following matrix components: `reproduction` (any transition to the first class); `survival` (surviving one time step and remaining in the same class); and `transition` (surviving one time step and moving to the next class).

aae.pop uses `masks` to select specific elements of the population matrix or the state vector (the vector of abundances in each class). All processes included in a population model are defined by a mask-function pair, with the mask specifying the cells to target and the function specifying what to do to these cells. Helper functions are included to define and combine common `masks`, including the `reproduction`, `survival`, and `transition` components described above, and general masks to select an entire population matrix (`all_cells`) or state vector (`all_classes`).

Building a population dynamics model

aae.pop is most easily illustrated with examples. A basic Leslie matrix with five age classes might be:

```
popmat <- rbind(
  c(0,    0,    2,    4,    7),  # reproduction from 3-5 year olds
  c(0.25, 0,    0,    0,    0),  # survival from age 1 to 2
  c(0,    0.45, 0,    0,    0),  # survival from age 2 to 3
  c(0,    0,    0.70, 0,    0),  # survival from age 3 to 4
  c(0,    0,    0,    0.85, 0)   # survival from age 4 to 5
)
```

The `dynamics` function converts this matrix into a `dynamics` object, which can be plotted (Figure 2) or passed to `simulate`:

```
# create a population dynamics object with a matrix and no other processes
popdyn <- dynamics(popmat)

# simulate 100 trajectories from this population dynamics object
#   (default is 50 updates, so 51 time slices when including initial
conditions)
sims <- simulate(popdyn, nsim = 100)
```

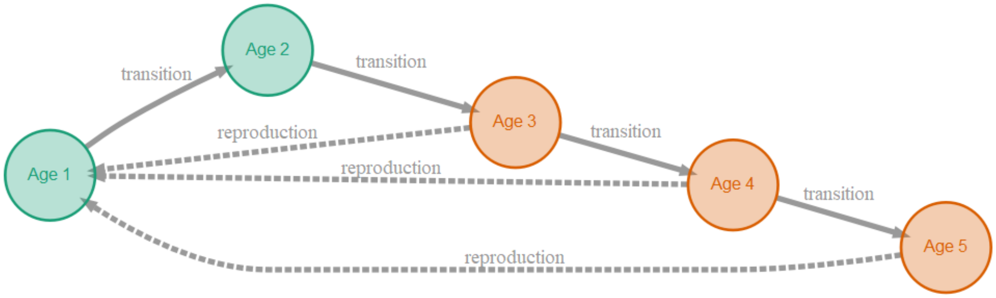

Figure 2: The structure of the example matrix population model.

The output of `simulate` is an array with dimensions of [`number of simulated trajectories`, `number of population classes`, `number of time steps`] (Figure 3). Several helper functions are included in **aae.pop** to calculate common summaries, including expected population statistics (`emps` for minimum values or `exps` for user-specified statistics), risk curves (`get_cdf`), and quasi-extinction probabilities (`pr_extinct`).

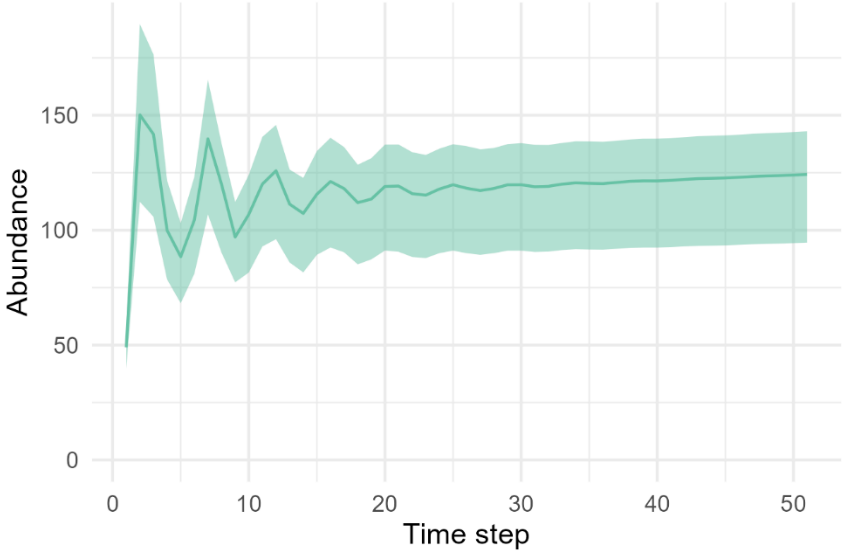

Figure 3: Example output from `simulate`. The solid line is the median population trajectory (over the `nsim = 100` simulated trajectories) and the shaded region bounds the 10th and 90th percentiles

Flexible specification of ecological processes

aae.pop introduces several process classes, all of which can be added to a `dynamics` object to modify simulated population dynamics. The key processes are shown in Figure 1. The following code illustrates these processes, starting with density dependence:

```r
# define a function for ceiling density dependence
#' @param x the vital rate being targeted
#' @param n the current population state vector
#' @param threshold the carrying capacity
dd_fn <- function(x, n, threshold) {

  # calculate total population size
  sum_n <- sum(n)

  # reduce the vital rate if n exceeds carrying capacity
  scaling_factor <- pmin(1, threshold / sum_n)

  # return rescaled vital rate
  scaling_factor * x

}

# use a mask to restrict dd to non-zero reproduction elements of popmat
dd_mask <- reproduction(popmat, dims = which(popmat[1, ] > 0))

# combine mask-function pair into a density_dependence object
dd <- density_dependence(
  masks = dd_mask,
  funs = dd_fn
)

# add this to the `dynamics` object
popdyn <- update(popdyn, dd)
```

The population `dynamics` object now includes density dependence, which requires a parameter `threshold` passed to `simulate`. The density dependence process can be updated by adjusting the `threshold` parameter (without altering `popdyn`) or by including an entirely different `density_dependence` term and updating the model (Figure 4):

```
# simulate from the population dynamics object, specifying a
#   threshold of 50 individuals
sims <- simulate(
  popdyn,
  nsim = 100,
  args = list(density_dependence = list(threshold = 50))
)

# simulate with a threshold of 20 individuals
sims_lower_threshold <- simulate(
  popdyn,
  nsim = 100,
  args = list(density_dependence = list(threshold = 20))
)

# specify a new dd model that sets reproduction to 0 when
#   threshold is exceeded
dd_fn2 <- function(x, n, threshold) {

  # calculate total population size
  sum_n <- sum(n)

  # set vital rate to zero if n exceeds carrying capacity
  scaling_factor <- ifelse(threshold / sum_n < 1, 0, 1)

  # return the rescaled vital rate
  scaling_factor * x

}
dd2 <- density_dependence(
  masks = dd_mask,
  funs = dd_fn2
)

# update the population dynamics model
popdyn <- update(popdyn, dd2)

# and simulate from this model
sims_new_dd <- simulate(
  popdyn,
  nsim = 100,
  args = list(density_dependence = list(threshold = 50))
)
```

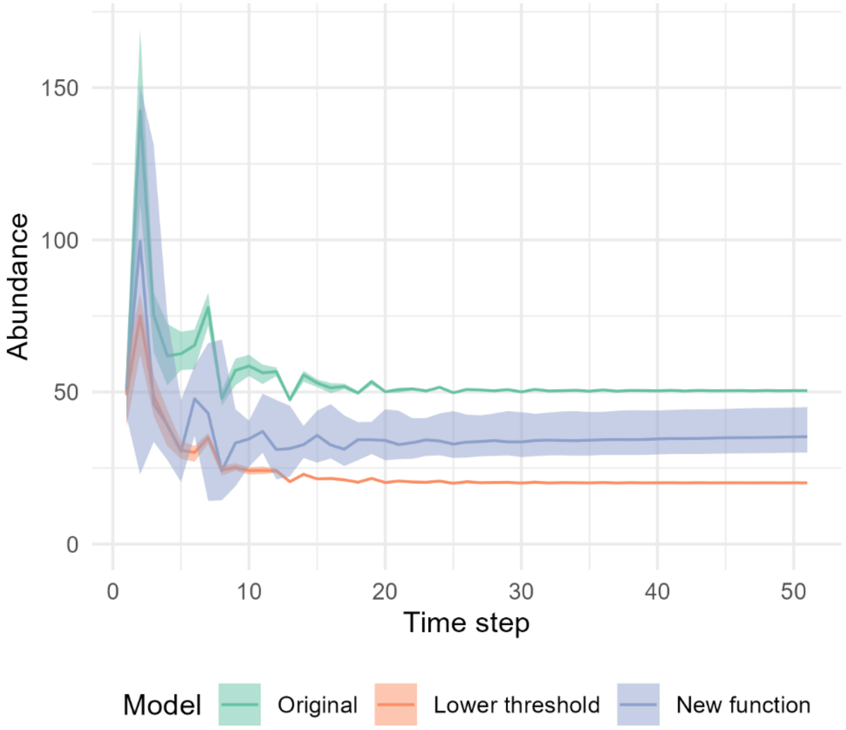

Figure 4: Example output from `simulate` *with two values of the density dependence threshold or an entirely different function. Solid lines are the median population trajectories and shaded regions bound the 10th and 90th percentiles.*

The mask-function pairing approach is used for all other processes, which allows many processes to be combined in a single model (Figure 5):

```
# define demographic stochasticity as Poisson variation on counts
demostoch <- demographic_stochasticity(
  masks = all_classes(popmat),
  funs = \(x) rpois(length(x), lambda = x)
)

# define environmental stochasticity as random 10% increases or
#   decreases in vital rates
envstoch <- environmental_stochasticity(
  masks = combine(
    reproduction(popmat, dims = 3:5), # dims restricts effects to classes 3-5
    transition(popmat)
  ),
  funs = \(x) runif(length(x), 0.9, 1.1) * x
)
```

```r
# add a covariate to modify reproduction based on the proportion
#   of available nesting habitat
covs <- covariates(
  masks = reproduction(popmat, dims = 3:5),
  funs = \(x, nests, ...) x * nests
)

# define the nesting covariate as a random value between 0.5 and 1
# Note: covariates define the number of time steps in the simulation
nyear <- 30
nesting <- runif(nyear, min = 0.5, max = 1.0)

# remove individuals each year due to predation, and add individuals
#   each year due to captive-bred releases
# Note: this illustrates stacking of mask-function pairs, which can be
#   used for any processes in aae.pop
add_remove <- add_remove_post(
  masks = list(
    all_classes(popmat, dims = 1:3),  # predation affects young classes
    all_classes(popmat, dims = 1)     # captive releases are in class 1
  ),
  funs = list(
    \(x, ...) pmax(0, x - 1),  # one individual eaten per year in each class
    \(x, ...) x + 15           # 15 captive-bred releases each year
  )
)

# combine this into a dynamics object
popdyn <- update(popdyn, demostoch, envstoch, covs, add_remove)

# and simulate
# Note: args are used to pass the dd threshold and nesting covariate
sims <- simulate(
  popdyn,
  nsim = 100,
  args = list(
    density_dependence = list(threshold = 100),
    covariates = format_covariates(nesting)
  )
)
```

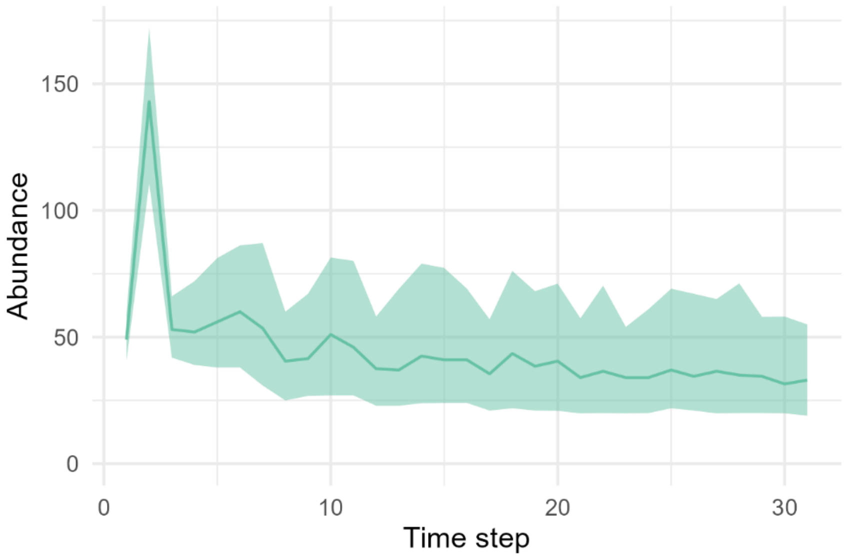

Figure 5: Example output from `simulate` *with multiple processes included in the* `dynamics` *object. The solid line is the median population trajectory and the shaded region bounds the 10th and 90th percentiles.*

Case studies

Case study 1: Modelling a highly dynamic fishery

Murray cod (*Maccullochella peelii*) are an endangered, long-lived freshwater fish species that occurs across the Murray-Darling Basin in south-eastern Australia. The species underwent declines in abundance throughout the 20th century, attributed to many factors, including a commercial fishery (now closed), changes in habitat availability, and widespread river regulation (Lintermans 2023).

This case study illustrates how **aae.pop** can combine relatively complex processes into a single MPM, and addresses the question of whether lingering effects of historical fishing might be responsible for low contemporary Murray cod population sizes. Specifically, **aae.pop** is used to model the impacts of the commercial fishery and the subsequent closure of this fishery on a generic Murray cod population.

The Murray cod population model is adapted from Todd *et al.* (2005) and includes 50 age classes, density dependence using a Beverton-Holt function, and size-dependent fishing selectivity. The model was used to simulate 100 replicate population trajectories (all code is included in the Supporting Information). Average vital rates are shown in Figure 6.

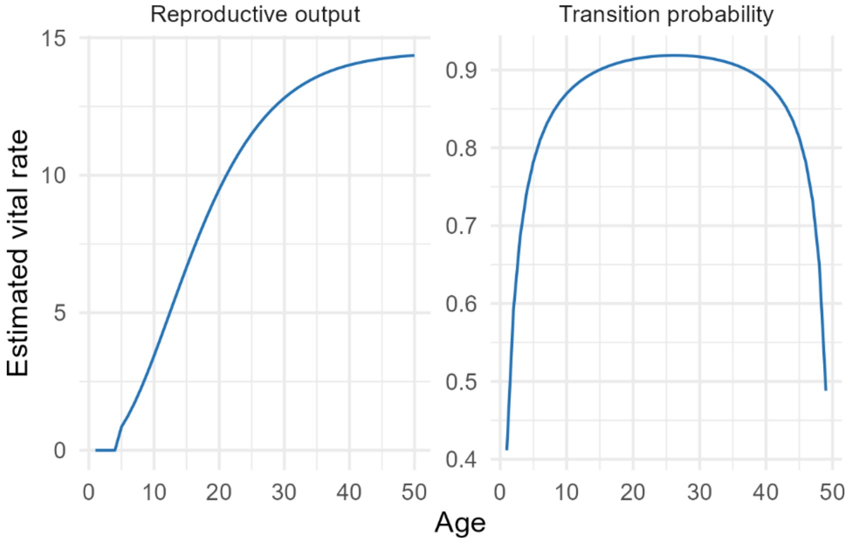

Figure 6: Average vital rates included in the Murray cod matrix population model. Reproductive output is defined for a pre-breeding census and includes survival of eggs, larvae, and fingerlings. Model parameters were derived from Todd et al. (2005).

```
# define the population matrix
mat <- fetch_matrix(Linf = 1300, k = 0.0674, t0 = 1.535)

# define a density dependence process that reduces reproduction
#   using the `bh` function
dd <- density_dependence(
  masks = reproduction(mat, dims = 5:50),
  funs = bh
)

# set the population carrying capacity (in kg)
carrying_capacity <- 20000
```

Stochastic environmental variation was included to capture variation in recruitment due to shifts between dry conditions and average or wetter conditions, and short periods of high mortality due to hypoxia caused by low flows and high temperatures or blackwater under extreme high flows:

```r
# define environmental stochasticity in reproduction
covs <- covariates(
  masks = list(reproduction(mat, dims = 5:50), transition(mat)),
  funs = list(
    \(x, climate, hypoxia) x * climate,
    \(x, climate, hypoxia) x * hypoxia
  )
)

# climate and hypoxia are defined here as 100-year, stochastic sequences
# Climate was modelled as an autoregressive (AR(1)) model with rho = 0.85,
#   rescaled to give values between +/- 10%
ntime <- 100
climate <- arima.sim(
  model = list(ar = 0.85), n = ntime, rand.gen = runif
)
climate <- as.numeric(climate)
climate <- climate - min(climate)
climate <- 0.9 + 0.2 * (climate) / max(climate)

# hypoxia was modelled as a random spike event, with an average of one
#    event every 20 years and magnitude of mortality between 40 % and 80 %
nspike <- rpois(1, lambda = ntime / 20)
idx <- sample(seq_len(ntime), size = nspike, replace = FALSE)
hypoxia <- rep(1, ntime)
hypoxia[idx] <- 1 - runif(nspike, min = 0.4, max = 0.8)
```

Fishing pressure was modelled as a total biomass take, assumed to be high for 50 years and then reduced for 50 years to represent low levels of recreational fishing following the cessation of commercial fishing. Fishing selectivity followed a logistic curve, with larger fish more likely to be removed than smaller fish:

```r
# define removals due to fishing, with selectivity governed within the
#   fishing_removals function
removals <- add_remove_post(
  masks = all_classes(mat, dims = 1:50),
  funs = fishing_removals
)
```

```r
# specify the rate of removals (in kg)
rate <- c(rep(2500, floor(ntime / 2)), rep(500, ntime - floor(ntime / 2)))
```

Initial conditions are required to simulate from an MPM. Here, initial conditions were based on the stable age distribution, with an additional burn-in period to remove transient effects.

Demographic stochasticity was implemented directly via the matrix update step (when the state vector is updated from one time step to the next) using the `update_binomial_leslie` updater in **aae.pop**. This updater uses a binomial distribution to apply demographic stochasticity to survival outcomes and a Poisson distribution for reproduction:

```r
# create and return the population dynamics object
dyn <- dynamics(mat, dd, covs, removals)

# set up initial conditions using the stable age distribution,
#    and simulating `nburnin` times to remove transient effects
init <- initialise_model(dyn, b0 = carrying_capacity, nburnin = 200)

# simulate from the model
# Note: `c()`is used to combine multiple arguments, which are then
#   passed to the relevant processes in this same order
sims <- simulate(
  dyn,
  nsim = 100,
  init = floor(init$init),
  args = list(
    density_dependence = list(carrying_capacity),
    add_remove_post = format_covariates(rate),
    environmental_stochasticity = c(
      format_covariates(climate),
      format_covariates(hypoxia)
    )
  ),
  options = list(update = update_binomial_leslie)
)
```

Simulated population trajectories suggest that Murray cod populations would be expected to have recovered in both abundance and biomass following the cessation of commercial fishing (Figure 7). The decline and subsequent recovery is much clearer in biomass trajectories than abundance trajectories, which stems from shifts in size structure as larger individuals are fished out of the population and replaced by smaller individuals. No decline is apparent in young-of-year recruits, which were not subjected to fishing pressure.

Although relatively simple model outputs are presented here, **aae.pop** returns full abundance trajectories in all population classes, which can be interrogated in many ways. One extension of this model might be to develop scenarios of the `climate` and `hypoxia` covariates, which could be used to compare population outcomes under different environmental futures (e.g., Hunter *et al.* 2010).

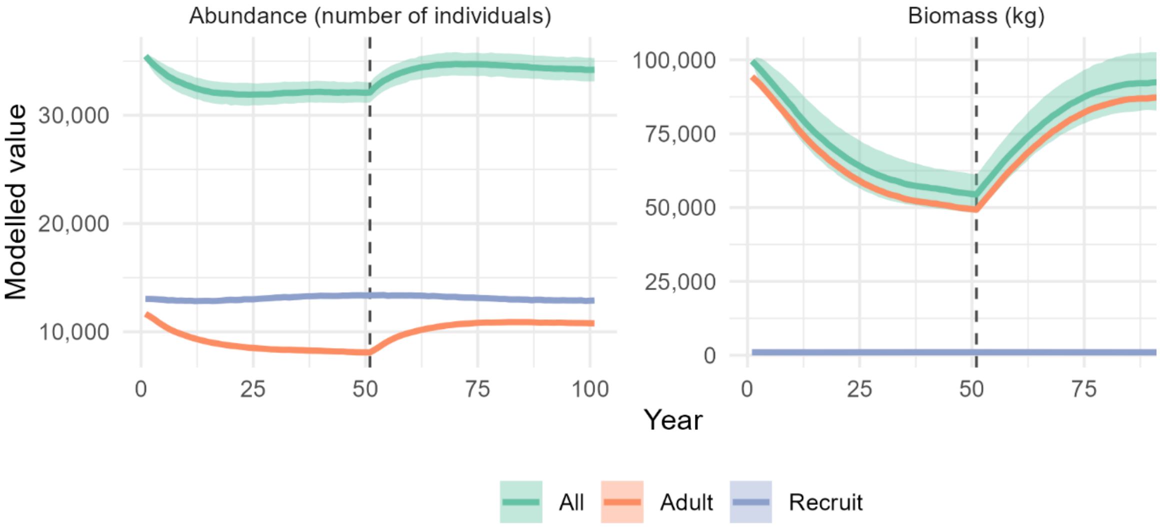

Figure 7: Simulated trajectories of Murray cod under a scenario of historical commercial fishing ceasing at year 50 (dashed line). Abundance and biomass trajectories are shown for the entire population, adults (5 years and older), and young-of-year recruits. Solid lines are median trajectories and shaded regions bound the 10th and 90th percentiles.

Case study 2: Modelling multiple, interacting species

This case study illustrates how **aae.pop** can be used to model multispecies dynamics, and includes environmental covariates, density dependence, environmental and demographic stochasticity, and pairwise interspecific interactions among three species. The case study uses the Murray cod population model developed above and adds models for the small-bodied Murray-Darling rainbowfish (*Melanotaenia fluviatilis*) and the invasive common carp (*Cyprinus carpio*). Natural environmental variability was assumed to affect recruitment and survival of all three species.

Pairwise species interactions modify vital rates for each population, similar to how intraspecific density dependence affects vital rates. In this case study, pairwise interactions were included to capture negative effects of common carp on both native species, particularly on smaller life stages, and predation by large Murray cod on smaller individuals of the other two species. Interactions for Murray cod and common carp were defined separately for young-of-year recruits, juveniles (1-4 years old), and adults (5 years and older). All classes of Murray-Darling rainbowfish were assumed to be small.

The three-species model was used to simulate 100 replicate population trajectories with and without species interactions for each species. This section presents results of the full simulation (code included in Supporting Information) but, for brevity, pseudo-code is used to illustrate how interspecific interactions can be included in **aae.pop**:

```r
# example function to specify a negative interaction between two species
#' @param x vital rates affected by species interaction
#' @param n population vector of the source species (i.e., the species
#'    that affects the other)
#' @param theta strength of interaction (larger = stronger interaction)
interaction_effect <- function(x, n, theta) {
  x * exp(-theta * sum(n))
}

# specify an interaction term for an aae.pop model, assuming
#   species1 and species2 are `dynamics` objects and that species2
#   affects species1 (but not vice versa)
interaction <- pairwise_interaction(
  target = species1,
  source = species2,
  masks = transition(species1$matrix, dim = affected_classes),
  funs = interaction_effect
)

# define the multispecies model, which contains `dynamics` objects
#   for all included species and their pairwise_interaction processes
msdyn <- multispecies(interaction)

# simulate with this model, passing `theta` as an argument to
#   the `interaction` process
sims <- simulate(
  msdyn,
  nsim = nsim,
  args = list(
    list(interaction = list(theta = theta)),
    list(interaction = list(theta = theta))
  )
)
```

Simulations of multispecies dynamics indicate that interspecific interactions have large effects on population trajectories (Figure 8). Older age classes of Murray cod and common carp benefit from pairwise interactions, whereas younger classes of both species and all classes of Murray-Darling rainbowfish are negatively affected by interactions. The effects of interspecific interactions are greater than the effects of environmental variation and, in most cases, considerably increase the variability in simulated population trajectories.

Figure 8: Simulated trajectories of Murray cod, common carp, and Murray-Darling rainbowfish under scenarios with and without interspecific interactions over a representative 16-year sequence of environmental variability. Solid lines are median trajectories and shaded regions bound the 10th and 90th percentiles. Trajectories are shown for young-of-year recruits, adults, and the entire population.

Discussion

The **aae.pop** package provides a flexible tool to build and simulate from MPMs for realistically complex ecological populations. Although equivalent simulations could be scripted directly or implemented in software such as RAMAS Metapop, **aae.pop** is open-source, freely available, and provides a standard framework and terminology for MPMs. The benefits of **aae.pop** will be most apparent when simulating repeatedly from an MPM with different inputs or ecological assumptions in each simulation. This situation occurs regularly in population viability analysis, and can be implemented in **aae.pop** with a single `dynamics` object and simulation-specific arguments or processes.

The use of `dynamics` objects and process classes (e.g., `density_dependence`) means that **aae.pop** models are highly portable, which supports model sharing and reproducibility. Existing databases of MPMs (e.g., COMPADRE, Salguero-Gómez *et al.* 2015) capture population vital rates but are less able to record details of model implementations, such as density dependence or covariate effects, especially when these processes are implemented algorithmically. An **aae.pop** `dynamics` object contains all the information needed to specify an MPM, and a full simulation can be reproduced from this object and a script containing the call to `simulate`. Importantly, the flexibility of **aae.pop** allows a user to update an existing model without having to code a model from scratch.

aae.pop is explicitly focused on applications of aspatial and spatially implicit MPMs, and complements existing packages for the analysis of matrix models (e.g., popdemo, mpmsim) and spatially explicit simulations (e.g., steps). Together, this suite of R packages provides a strong foundation for reproducible models of population dynamics and their applications to real-world ecological challenges. What remains is to ensure that these MPMs are reliable and free from errors (Kendall *et al.* 2019), a task well supported by the portability of MPMs developed in **aae.pop**.

Acknowledgements

I thank Charles Todd, Zeb Tonkin, Henry Wootton, Jim Thomson, and Rachel Leihy for discussions on the methods, models, and case studies presented here. Rachel Leihy provided feedback on a draft manuscript. Joslin Moore, Cindy Hauser, Matt Bruce, Michael Scroggie, Simone Stevens, Tracey Hollings, Tracey Regan, and Carlo Pacioni provided feedback on the **aae.pop** software package. This work was funded by the Department of Energy, Environment and Climate Action (DEECA). I thank Anne Buchan, Billy Geary, Ella Kelly, Rob Hale, Paul Reich, and Rachel Austin for facilitating this funding.

Funding information

This work was supported by the Victorian Department of Energy, Environment and Climate Action (DEECA).

Data availability statement

The aae.pop R package is publicly available on CRAN (DOI: 10.32614/CRAN.package.aae.pop). Code to recreate the analyses and examples presented in the preprint is available at Zenodo (DOI: 10.5281/zenodo.19901285).

References

Akçakaya, H., McCarthy, M.A. & Pearce, J.L. (1995). Linking landscape data with population viability analysis: Management options for the helmeted honeyeater *Lichenostomus melanops cassidix*. *Biological Conservation*, **73**: 169–176. https://doi.org/10.1016/0006-3207(95)90045-4

Boyce, M.S. (1992). Population viability analysis. *Annual Review of Ecology, Evolution, and Systematics*, **23**, 481–497. https://doi.org/10.1146/annurev.es.23.110192.002405

Caswell, H. (2001). *Matrix population models (2nd ed.)*. Sinauer Associates, Inc.

Coste, C.F.D. & Pavard, S. (2020). Analysis of a multitrait population projection matrix reveals the evolutionary and demographic effects of a life history trade-off. *Ecological Modelling*, **418**, 108915. https://doi.org/10.1016/j.ecolmodel.2019.108915

Ezard, T.H.G., Bullock, J.M., Dalgleish, H.J., Millon, A., Pelletier, F., Ozgul, A. & Koons, D.N. (2010). Matrix models for a changeable world: The importance of transient dynamics in population management. *Journal of Applied Ecology*, **47**, 515–523. https://doi.org/10.1111/j.1365-2664.2010.01801.x

Hunter, C.M., Caswell, H., Runge, M.C., Regehr, E.V., Amstrup, S.C. & Stirling, I. (2010). Climate change threatens polar bear populations: A stochastic demographic analysis. *Ecology*, **91**, 2883–2897. https://doi.org/10.1890/09-1641.1

Jones, O.R. (2025). mpmsim: An R package for simulating matrix population models. *Methods in Ecology and Evolution*, **16**, 904–911. https://doi.org/10.1111/2041-210X.70030

Kendall, B.E., Fujiwara, M., Diaz-Lopez, J., Schneider, S., Voigt, J. & Wiesner, S. (2019). Persistent problems in the construction of matrix population models. *Ecological Modelling*, **406**, 33–43. https://doi.org/10.1016/j.ecolmodel.2019.03.011

Lintermans, M. (2023). *Fishes of the Murray-Darling Basin (2nd ed.)*. Australian River Restoration Centre.

Salguero-Gómez, R., Jones, O.R., Archer, C.R., Buckley, Y.M., Che-Castaldo, J., Caswell, H., Hodgson, D., Scheuerlein, A., Conde, D.A., Brinks, E. & others. (2015). The COMPADRE plant matrix database: An open online repository for plant demography. *Journal of Ecology*, **103**, 202–218. https://doi.org/10.1111/1365-2745.12334

Stott, I., Hodgson, D.J. & Townley, S. (2012). Popdemo: An R package for population demography using projection matrix analysis. *Methods in Ecology and Evolution*, **3**, 797–802. https://doi.org/10.1111/j.2041-210X.2012.00222.x

Stubben, C. & Milligan, B. (2007). Estimating and analyzing demographic models using the popbio package in R. *Journal of Statistical Software*, **22**, 1–23. https://doi.org/10.18637/jss.v022.i11

Todd, C.R., Ryan, T., Nicol, S.J. & Bearlin, A.R. (2005). The impact of cold water releases on the critical period of post-spawning survival and its implications for Murray cod (*Maccullochella peelii peelii*): A case study of the Mitta Mitta River, southeastern Australia. *River Research and Applications*, **21**, 1035–1052. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/rra.873

Visintin, C., Briscoe, N.J., Woolley, S.N.C., Lentini, P.E., Tingley, R., Wintle, B.A. & Golding, N. (2020). steps: Software for spatially and temporally explicit population simulations. *Methods in Ecology and Evolution*, **11**, 596–603. https://doi.org/10.1111/2041-210X.13354