Received: 14 October 2021 | Accepted: 30 November 2021
DOI: 10.1111/2041-210X.13792

Methods in Ecology and Evolution
BRITISH ECOLOGICAL SOCIETY

APPLICATION

Rcompadre and Rage—Two R packages to facilitate the use of the COMPADRE and COMADRE databases and calculation of life-history traits from matrix population models

Owen R. Jones[1] | Patrick Barks[1] | Iain Stott[1,2] | Tamora D. James[3] | Sam Levin[4] | William K. Petry[5] | Pol Capdevila[6] | Judy Che-Castaldo[7] | John Jackson[1] | Gesa Römer[1] | Caroline Schuette[7] | Chelsea C. Thomas[7] | Roberto Salguero-Gómez[8,9]

[1]University of Southern Denmark, Odense C, Denmark; [2]School of Life Sciences, University of Lincoln, Lincoln, UK; [3]Department of Animal and Plant Sciences, University of Sheffield, Western Bank, Sheffield, UK; [4]Helmholtz-Centre for Environmental Research—UFZ, Martin Luther University Halle-Wittenberg, Leipzig, Germany; [5]Department of Plant and Microbial Biology, North Carolina State University, Raleigh, North Carolina, USA; [6]School of Biological Sciences, University of Bristol, Bristol, UK; [7]Alexander Center for Applied Population Biology, Conservation & Science Department, Lincoln Park Zoo, Chicago, Illinois, USA; [8]Department of Zoology, University of Oxford, Oxford, UK and [9]Max Planck Institute for Demographic Research, Rostock, Germany

Correspondence
Owen R. Jones
Email: jones@biology.sdu.dk

Funding information
Natural Environment Research Council, Grant/Award Number: NE/M018458/1; Division of Biological Infrastructure, Grant/Award Number: DBI-1661342; Det Frie Forskningsråd, Grant/Award Number: DFF-6108-00467; Max Planck Institute for Demographic Research

Handling Editor: Timothée Poisot

Abstract

1. Matrix population models (MPMs) are an important tool for biologists seeking to understand the causes and consequences of variation in vital rates (e.g. survival, reproduction) across life cycles. Empirical MPMs describe the age- or stage-structured demography of organisms and usually represent the life history of a population during a particular time frame at a specific geographical location.
2. The COMPADRE Plant Matrix Database and COMADRE Animal Matrix Database are the most extensive resources for MPM data, collectively containing >12,000 individual projection matrices for >1,100 species globally. Although these databases represent an unparalleled resource for researchers, land managers and educators, the current computational tools available to answer questions with MPMs impose significant barriers to potential COM(P)ADRE database users by requiring advanced knowledge to handle diverse data structures and program custom analysis functions.
3. To close this knowledge gap, we present two interrelated R packages designed to (a) facilitate the use of these databases by providing functions to acquire, quality control and manage both the MPM data contained in COMPADRE and COMADRE, and a user's own MPM data (**Rcompadre**) and (b) present a range of functions to calculate life-history traits from MPMs in support of ecological and evolutionary analyses (**Rage**). We provide examples to illustrate the use of both.
4. **Rcompadre** and **Rage** will facilitate demographic analyses using MPM data and contribute to the improved replicability of studies using these data. We hope that this new functionality will allow researchers, land managers and educators

to unlock the potential behind the thousands of MPMs and ancillary metadata stored in the COMPADRE and COMADRE matrix databases, and in their own MPM data.

KEYWORDS
ageing, age-structured population model, life-history strategy, matrix population model, population dynamics, population projection model, stage-structured population model

1 | INTRODUCTION

Matrix population models (MPMs, hereafter) have become a commonplace tool for ecologists, evolutionary biologists and conservation biologists seeking to understand how variation in vital rates (e.g. survival, development, reproduction, recruitment, etc.) in the life cycle varies geographically and across species. MPMs describe population dynamics based on stage- or age-specific vital rates in the population of interest over their life cycle (Caswell, 2001). Outputs derived from MPMs include population growth rates (Caswell, 2001), key life-history traits (Caswell, 2001) and vital rate sensitivities (de Kroon et al., 1986, 2000). These outputs each have a well-understood biological interpretation, which allows comparison of MPM-derived population and life-history metrics, and thus demography across the diversity of life on Earth, from moss (e.g. Okland, 1995) to monkeys (e.g. Morris et al., 2011) to microbes (e.g. Jouvet et al., 2018), and in myriad ecoregions.

Since the introduction of MPMs in the 1940s (Leslie, 1945, 1948), researchers have published thousands of MPMs for thousands of species. Our team has been digitising these MPMs into centralised databases for plants (the COMPADRE Plant Matrix Database: Salguero-Gómez et al., 2015) and animals (the COMADRE Animal Matrix Database: Salguero-Gómez et al., 2016). These twin databases now contain more than 12,000 MPMs for more than 1,100 species (COMPADRE: 8,708 matrices for 757 species; COMADRE: 3,317 matrices for 415 species, as of September 2021) and are regularly augmented with newly published and newly digitised records. The databases, their history and the rationale behind the data organisation are described in Salguero-Gómez et al. (2015) and Salguero-Gómez et al. (2016), respectively.

COMPADRE and COMADRE store and provide MPMs and their associated metadata in a hierarchical structure that, while efficient for distribution, can be both a barrier to use and an entry point for user errors. The primary component of MPMs are the two-dimensional, square projection matrices, and the size of these matrices can vary widely across species and studies. Moreover, most projection matrices (**A**) in the databases are partitioned into their three constituent process-based submatrices such that $\mathbf{A} = \mathbf{U} + \mathbf{F} + \mathbf{C}$. Here, submatrix **U** describes transitions related to survival and growth/development, submatrix **F** describes sexual reproduction and submatrix **C** describes clonal reproduction. Thus, in most cases, each MPM is represented by these four matrices (**A**, the main projection matrix and the submatrices **U**, **F** and **C**) alongside information about the life cycle stages used in the MPM. In the majority of cases, the projection interval (time step) for the MPM is 1 year, but this can vary considerably depending on the life history of the organism concerned (e.g. 5-year intervals are common in tree MPMs). Each MPM in the databases is also associated with over 40 metadata variables extracted from its parent original work(s) (e.g. stage definitions, projection time steps, citation, taxonomy, geography, etc., detailed in Salguero-Gómez et al., 2016, 2015). This nested structure allows for higher digitisation fidelity and distribution efficiency, but also means that the dataset cannot be imported by ordinary spreadsheet software, such as Excel, which accommodate only rectangular (or 'flat') data structures. Both of the most common tools for working with MPMs, the R statistical programming language (R Core Team, 2021) and Matlab (Matlab, 2010), readily accept hierarchical data structures. However, users must have a familiarity with handling a range of nested object classes to organise the databases to suit their needs (e.g. '*subset to only primates*' or '*subset to only species from tropical ecoregions*'). The higher dimensionality can increase the risk of errors, such as using the wrong data dimension, even for experienced users.

The R package ecosystem provides a wide range of tools for analysing population dynamics from MPMs within individual populations. For example, **popdemo** (Stott et al., 2012) focuses on the calculation of metrics related to transient population dynamics and transfer function analyses; **popbio** (Stubben & Milligan, 2007) provides functions to accomplish many (but not all) of the analyses found in the textbooks of Caswell (2001) and Morris and Doak (2002), such as the calculation of eigen properties (i.e. the asymptotic population growth rate, stable stage structure and reproductive values) or sensitivities and elasticities; **Rramas** (de la Cruz Rot, 2019) provides tools for making population projections and conducting population viability analyses from MPM data; and **lefko3** (Shefferson et al., 2021) provides tools that allow the inclusion of information on individual histories, which could influence population dynamics, into MPM analyses (see Ehrlén, 2000). However, the tools for life-history analysis provided by these existing packages are more limited, with among the most notable absence being important life-history metrics based on age-from-stage calculations. Researchers that wanted to make such calculations (e.g. measures of senescence, longevity or age at maturity) have needed to write their own code based on published

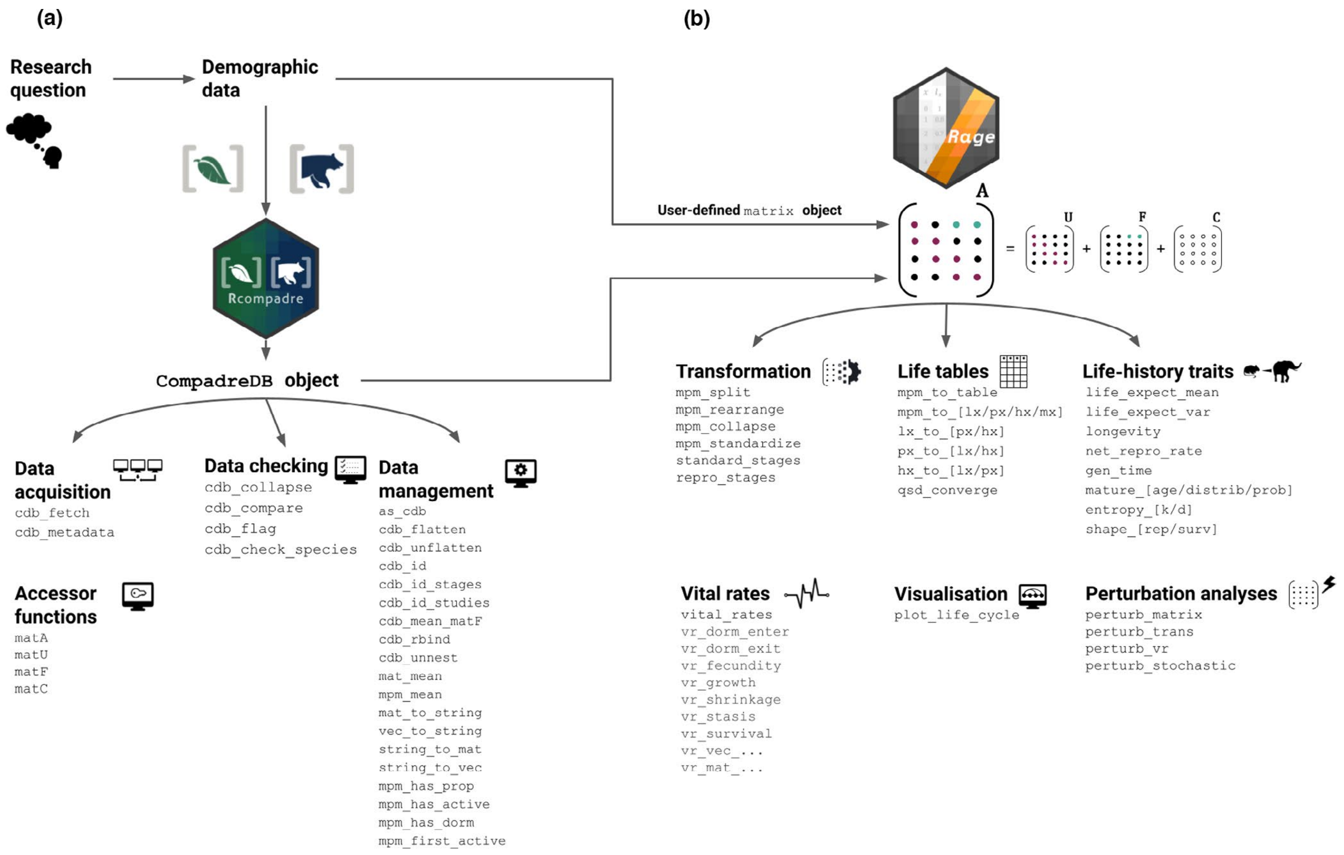

FIGURE 1 Workflow of using **Rcompadre** and **Rage** for ecological and evolutionary analyses of matrix population model data. (a) Once the author(s) have identified the research question, demographic data in the format of MPMs can be accessed from the COMPADRE and/or COMADRE databases via the **Rcompadre** R package. This package allows for the online acquisition, checking (according to data needs) and management of the `CompadreDB` data object (e.g. using `cdb_fetch` to download the data and `cdb_flag` and `filter/subset` to produce a dataset for analysis). (b) The filtered data (or other user-provided MPM data) can be then migrated for calculations of life-history traits with **Rage** (alternatively, these can be done directly on MPMs provided by the author). The families of functions archived in **Rage** include transformation (e.g. `mpm_collapse`), creation of life tables (e.g. `mpm_to_lx`), derivation of life-history traits (e.g. `longevity`), calculation of vital rates (e.g. using `vital_rates` to calculate average survival, reproduction, development, etc.), visualisation of life cycles (e.g. `plot_life_cycle`) and perturbation analyses (e.g. `perturb_stochastic`)

equations in mathematics heavy work, which has been a barrier to the broader adoption of these methods. Moreover, these life-history metrics are often most meaningful in analyses across many populations or species. The existing packages provide little support for the large hierarchical data structures needed to apply analyses to hundreds or thousands of MPMs that may underlie a single comparative or macroecological analysis.

Here, we introduce two R packages that enable users to construct robust MPM analysis workflows to answer questions from single populations to across the tree of life. The first package, **Rcompadre**, is designed to facilitate acquisition, quality control and management of the rich, hierarchical MPM data in COMPADRE and COMADRE. For example, this package includes tools to filter (subset) the databases based on metadata archived in these resources (e.g. by ecoregion, by taxonomic group). In addition to 'base' style R syntax for these tasks, **Rcompadre** integrates **tidyverse** (Wickham et al., 2019) functionality to improve usability. The second package, **Rage**, builds on the enhanced data accessibility provided by **Rcompadre** by providing analysis pipeline support for arbitrarily large numbers of MPMs and the calculation of life-history traits needed to support comparative analyses on this scale. These life-history traits include life tables, mean life expectancy, generation time, among several others.

We showcase downloading, subsetting and preparing MPM data for a broad comparative analysis using publicly accessible data retrieved with **Rcompadre** (Box 1). We then illustrate an application of **Rage** to calculate ecologically and evolutionarily relevant metrics to test hypotheses related to life-history theory at broad taxonomic scale. In doing so, we demonstrate the functional integration of **Rcompadre** and **Rage** and how investigators can use them in tandem to design workflows (Figure 1) to answer their own questions in ecology, evolution and conservation biology.

2 | Rcompadre

Rcompadre contains functions to facilitate downloading and using MPMs alongside their metadata from the COMPADRE and

BOX 1 Using Rcompadre to download and prepare MPM data for analysis

In the following example, we illustrate the use of Rcompadre to carry out typical data download and preparation tasks for an analysis relevant to comparative population dynamics research. Specifically, we aim towards an analysis of mammalian life span and its relationship with generation time (continued in Box 2).

After loading the required packages, we download the COMADRE data and conduct some basic checks of the matrices. We then filter the dataset to include only mammals, to include no missing values in the **U** matrix, and to ensure that the **U** and **F** matrices are not filled entirely with zero values, nor that columns of the **U** matrix sum to 0. We further filter the data to ensure that the projection interval is 1 year. Finally, we can plot the geographical distribution of these data using tools from the ggplot2 and maps packages (Figure 2).

```
 # Load packages
library(Rcompadre)
library(tidyverse)
 # Fetch data, and conduct basic checks
comadre <- cdb_fetch("comadre", flag = TRUE)
 # Filter for mammals, split matrices, NA/0 values in U and F matrices and a
 # projection interval of 1
mammals <- comadre %>%
 filter(Class == "Mammalia") %>%
 filter(MatrixSplit == "Divided") %>%
 filter(
 check_NA_U == FALSE, check_zero_U == FALSE,
 check_zero_F == FALSE, check_zero_U_colsum == FALSE
 ) %>%
 filter(ProjectionInterval == 1)
 # Plot geographic distribution
ggplot(mammals, aes(x = Lon, y = Lat)) +
 borders(database = "world", fill = "grey80", col = NA) +
 geom_point(alpha = 0.4, color = "#E69F00") +
 scale_x_continuous(breaks = seq(-180, 180, 90), expand = c(0, 0)) +
 scale_y_continuous(expand = c(0, 0)) +
 labs(x = "Longitude", y = "Latitude") +
 theme_minimal()
```

COMADRE databases (Figure 1a). A central feature of this package is the definition of a new object class, `CompadreDB`, which allows R functions that are already familiar to users (e.g. `head` or **tidyverse** verbs) to be augmented with 'methods' that ensure that they appropriately handle the structure of MPM data from the COM(P)ADRE databases. In addition to improving user-friendliness, the class definition provides a pathway for extending the compatibility of COM(P)ADRE data to other existing or future R functions. Briefly, the structure of `CompadreDB` objects uses the S4 systems[1] with two slots: (a) the `data` slot, which contains a `tibble`-style data frame (Wickham & Grolemund, 2016) with a list column of MPMs and vector columns of metadata and (b) the `version` slot which contains database version information for reproducibility, including the version number, date created and a link to the database user agreement. In addition, we have created the `CompadreMat` class, which formally defines how MPMs are represented in a `CompadreDB` object. Here too, the use of an explicit class definition has allowed us to define how the data contained in the object will respond to familiar R functions. For example, users can access and replace columns of data using the standard `x$name` and `x$name <- value` methods, respectively. In addition, we provide the functionality to access the matrix data directly, for example, using the functions `matA` or `matU` to access all **A** matrices or **U** submatrices in the database as a list. This functionality is particularly convenient if the user wishes to apply functions to a large set of MPMs, as one would do in comparative and macroecological analyses (e.g. see recent studies by Coutts et al. (2016), Capdevila et al. (2020), Takada and Kawai (2020), James et al. (2020),

---

[1] R includes significant support for object-oriented programming, and the S4 system is one of R's systems for defining object classes. It is a stricter, less flexible system than R's base system (S3) but has the advantage of enhancing consistency in how objects are defined and handled, and in the ease with which data can be accessed from nested objects. The details are far beyond the scope of this article, but see Wickham (2019) for fuller coverage.

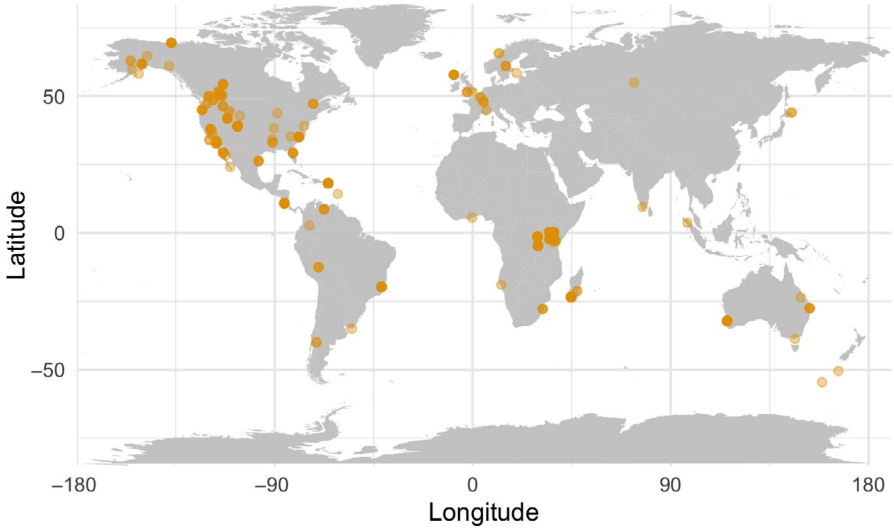

FIGURE 2 The spatial extent of data in the subset of mammal data used in our example analysis. Note that 186 of the matrices for mammals in our set (~27%) lack associated spatial information

Jones et al. (2020) and Healy et al. (2019)). In addition to 'base' R functions, many data analysis workflows make use of functions in the **tidyverse** family of packages (Wickham et al., 2019). Our package includes 'tidy' methods for `CompadreDB` objects, allowing users to `filter`, `arrange`, `mutate`, `select`, `summarise`, `rename` and `join` COM(P)ADRE data to answer their study questions efficiently and at scale. The provision of these **tidyverse** methods also means that **Rcompadre** benefits from the piping (e.g. `%>%`) functionality of **magrittr** and more recently in base R (`|>`, in v.4.1.0 and later). Examples of how this functionality can streamline the human readability of workflows can be found in the vignettes at the package development pages.

In addition to a wide range of method-based support of existing R functions, **Rcompadre** provides functions for additional workflow tasks that follow the naming pattern of `cdb_` (pronounced 'compadre database') followed by a meaningful verb. For example, `cdb_fetch` retrieves COM(P)ADRE data of the current or any previous database version from the web as a `CompadreDB` object, and `cdb_compare` reports the differences between any pair of `CompadreDB` objects. Table 1 summarises the most important **Rcompadre** functions, and full documentation of all functions is provided in the package manual.

2.1 | Data management and checking

The COM(P)ADRE databases include metadata associated with each MPM including taxonomic information, geolocation and details of the source publication (see the User Guide at www.compadre-db.org or Salguero-Gómez et al., 2016, 2015 for full metadata documentation). When working with these data via Rcompadre, we can see the richness of the metadata with R's `names` function and users can use any of these metadata columns to filter the database prior to analysis. The projection matrices themselves are contained in a list column called `mat`, where each element includes a list of the four matrices: **A** and the submatrices **U**, **F** and **C** (see above). The list also provides information on matrix stage definitions. All other columns of the COMADRE database object are ordinary vectors.

Not all COM(P)ADRE data will meet the inclusion criteria for a particular analysis. **Rcompadre** includes several general functions for checking the data that use the quality control flags generated when MPMs are digitised and checked before addition to the databases. These data checks are accessed through **Rcompadre** using the `cdb_flag` function. This function, which can be implemented as a stand-alone function or during data retrieval by `cdb_fetch`, adds logical metadata columns to the provided `CompadreDB` object which can be used for data filtering (see `?cdb_flag` for details of the available data property checks). For example, a minority of studies published only the main projection matrix, **A**, thereby preventing its decomposition into the **U**, **F** and **C** submatrices which may preclude certain demographic analyses. Matrices may also have missing (`NA`) values where a transition was not estimated. Other potential pitfalls flagged by this function include matrices that are singular (non-invertible), non-ergodic (where initial stage structure can influence asymptotic population growth rate), reducible (where the associated life cycle graph does not contain all necessary transition rates to enable pathways from all stages to all other stages) or non-primitive (Caswell, 2001; Stott et al., 2010). Depending on the desired downstream analyses, researchers may need to `filter` the database based on one or more of these flag columns.

The quality checks performed by `cdb_flag` cannot anticipate all potential inclusion criteria, and we strongly encourage investigators to perform additional checks that may be necessary to determine the suitability of a MPM record for their analysis. The existing metadata columns associated with each MPM contain a wealth of useful information to this end. For example, the interpretation of many metrics derived from MPMs depends on the projection interval (`ProjectionInterval`). We advise users to `filter` on this column to a common projection interval prior to analysis or to correct analysis outputs to the same temporal units. An analysis may also require delineating MPM records that use post- versus

TABLE 1 The functions in **Rcompadre** are grouped into four categories: Data acquisition, Data checking, Data management and Accessor functions. We outline the most important functions here, with a brief description. Users should consult the package documentation for a full description of named functions (e.g. `?cdb _ fetch`) and to see a full list of functions

Category	Function	Description
Data acquisition	`cdb _ fetch()`	Downloads the current version of the COMPADRE or COMADRE databases, or loads a local database file
	`cdb _ metadata()`	Extracts a `tibble` with only metadata from a `CompadreDB` object
Data checking	`cdb _ collapse()`	Collapses a `CompadreDB` object by averaging projection matrices over levels of one or more grouping variables
	`cdb _ compare()`	Compares two versions or subsets of `CompadreDB` objects
	`cdb _ flag()`	Flags potential problems with projection matrices within a `CompadreDB` object, such as missing values, singular **U** submatrices, non-ergodicity, non-irreducibility, primitivity, etc. (see Iain Stott et al., 2012)
	`cdb _ check _ species()`	Checks for specific species in a `CompadreDB` object
Data management	`as _ cdb()`	Generates an S4 `CompadreDB` object from S3 formatted data
	`cdb _ flatten()`	Converts a `CompadreDB` object into a flat data frame with projection matrices and vectors stored in string representation
	`cdb _ unflatten()`	Converts a flattened data frame back into a `CompadreDB` object
	`cdb _ id()`	Creates a vector of integer identifiers corresponding to unique combinations of a given set of columns
	`cdb _ id _ stages()`	Creates a vector of integer identifiers corresponding to unique combinations of a species and matrix stage class definitions
	`cdb _ id _ studies()`	Creates a vector of integer identifiers corresponding to unique combinations of publication metadata
	`cdb _ mean _ matF()`	Calculates a population-specific mean fecundity submatrix (**F**) for each set of projection matrices in a `CompadreDB` object
	`cdb _ rbind()`	Merges two `CompadreDB` objects using a row-bind of the data slots
	`cdb _ unnest()`	Unnests a `CompadreDB` object by spreading the nested components of `CompadreMat` into separate columns
	`mat _ mean(), mpm _ mean()`	Calculates an element-wise mean over a list of projection matrices or `CompadreMat` objects
	`mat _ to _ string(), vec _ to _ string(), string _ to _ mat(), string _ to _ vec()`	Converts vectors or square numeric matrices to and from string representation
	`mpm _ has _ prop(), mpm _ has _ active(), mpm _ has _ dorm()`	Extracts stage-class information (e.g. propagule, dormant and active stages) from a `CompadreMat` or `CompadreDB` object
	`mpm _ first _ active()`	Extracts the integer index of the first active (i.e. non-dormant, non-seedbank) stage class in a `CompadreMat` or `CompadreDB` object
Accessor functions	`matA(), matU(), matF(), matC()`	Extracts full projection matrix (**A**), or the survival (**U**), sexual reproduction (**F**) or clonal reproduction (**C**) submatrices from a `CompadreMat` or `CompadreDB` object

pre-reproductive census models. Although both databases have a metadata field that reports this information (`CensusType`), it is often not reported in original publications and thus COM(P)ADRE includes records with incomplete metadata. Users may therefore need to carefully consider the source publication (e.g. retrieved using the `DOI _ ISBN` and `AdditionalSource` column metadata) or contact the original authors to determine suitability.

Finally, **Rcompadre** includes a function, `cdb _ build _ cdb`, which allows users to access the full functionality of **Rcompadre** for their own data by constructing valid `CompadreDB` objects from user-supplied lists of matrices, (optional) stage information and an accompanying data frame of metadata. Furthermore, we provide a way for users to augment COM(P)ADRE with a `CompadreDB` object containing their own data using the function `cdb _ rbind`. This nimble data extensibility ensures the continued utility of **Rcompadre**'s suite of workflow tools without dependency on externally maintained data.

In Box 1, we illustrate the use of **Rcompadre** to download, check and filter the COMADRE database (animal MPMs) in preparation for a later analysis of mammal life span using **Rage**. Vignettes at

TABLE 2 The functions in Rage are grouped into six categories: Life-history traits, Life tables, Vital rates, Perturbation analyses, MPM transformation and Visualisation. We outline the most important functions here with a brief description. Users should consult the package documentation for a full description of named functions (e.g. `?life_expect_mean`) and to see a complete list of functions

Category	Function	Description
Life history traits	`life_expect_mean()`, `life_expect_var()`	Applies Markov chain approaches to obtain the mean and/or variance of life expectancy from a matrix population model
	`longevity()`	Calculates the age at which survivorship falls below some critical proportion from a matrix population model (see SI in Jones et al., 2014)
	`net_repro_rate()`	Calculates net reproductive value (*R0*) from a matrix population model
	`gen_time()`	Calculates generation time from a matrix population model
	`mature_age()`, `mature_distrib()`, `mature_prob()`	Calculates the mean age at first reproduction, the stage distribution of individuals achieving reproductive maturity and the probability of achieving reproductive maturity using Markov chain approaches
	`entropy_d()`	Calculates Demetrius' entropy (Demetrius, 1978) from vectors of age-specific survivorship (l_x) and fecundity (m_x)
	`entropy_k()`	Calculates Keyfitz's entropy (Keyfitz & Caswell, 2005) from a vector of age-specific survivorship (l_x)
	`shape_rep()`	Calculates a 'shape' value for distribution of reproduction over age (Baudisch & Stott, 2019)
	`shape_surv()`	Calculates a 'shape' value for survival lifespan inequality (Baudisch, 2011)
Life tables	`mpm_to_table()`	Generates a life table from a matrix population model using age-from-stage decomposition methods (Caswell, 2001; Cochran & Ellner, 1992)
	`mpm_to_hx()`, `mpm_to_lx()`, `mpm_to_mx()`, `mpm_to_px()`	Calculates mortality hazard (h_x), age-specific survivorship (l_x), reproduction (m_x), and survival probability (p_x) from a matrix population model using age-from-stage decomposition methods
	`lx_to_px()`, `lx_to_hx()`, `px_to_lx()`, `px_to_hx()`, `hx_to_lx()`, `hx_to_px()`	Converts between vectors of age-specific survivorship (l_x), survival probability (p_x), and mortality hazard (h_x)
	`qsd_converge()`	Calculates the time for a cohort projected with a matrix population model to reach a defined quasi-stationary stage distribution (see SI in Jones et al., 2014)
Vital rates	`vitalRates()`	Derives the mean vital rates for a matrix population model
	`vr_dorm_enter()`, `vr_dorm_exit()`, `vr_fecundity()`, `vr_growth()`, `vr_shrinkage()`, `vr_stasis()`, `vr_survival()`	Derives mean vital rates of survival, growth (or development), shrinkage (or de-development), stasis, dormancy, or reproduction from a matrix population model, by averaging across stage classes
	`vr_vec_dorm_enter()`, `vr_vec_dorm_exit()`, `vr_vec_growth()`, `vr_vec_reproduction()`, `vr_vec_shrinkage()`, `vr_vec_stasis()`, `vr_vec_survival()`	Derives vectors of stage-specific vital rates of survival, growth, shrinkage, stasis, dormancy, or reproduction from a matrix population model
	`vr_mat_R()`, `vr_mat_U()`	Derives survival-independent vital rates for growth, stasis, shrinkage, and reproduction
Perturbation analyses	`perturb_matrix()`	Perturbation analysis of an emerging demographic property (e.g., population growth rate, damping ratio) with respect to changes on matrix elements
	`perturb_trans()`	Perturbation analysis of transition types within a matrix population model
	`perturb_vr()`	Perturbation analysis of underlying vital rates (Franco & Silvertown, 2004) in a matrix population model
	`perturb_stochastic()`	Perturbation analysis of an emerging demographic property (e.g. population growth rate, damping ratio) with respect to changes on matrix elements

TABLE 2 (Continued)

Category	Function	Description
MPM transformation	`mpm_collapse()`	Collapses a matrix population model to a smaller number of stages using weighted averages (Salguero-Gómez & Plotkin, 2010)
	`mpm_rearrange()`	Rearranges the stages of a matrix population model to segregate reproductive and non-reproductive stages
	`mpm_split()`	Converts a matrix population model into survival (**U**), fecundity (**F**) and clonal (**C**) matrices
	`mpm_standardize()`	Transforms a matrix population model to a standardised set of stage classes
	`repro_stages()`	Identifies which stages in a matrix population model are reproductive
	`standard_stages()`	Identifies the stages of a matrix population model that correspond to different parts of the reproductive life cycle
Visualisation	`plot_life_cycle()`	Plots a life cycle diagram from a matrix population model

the **Rcompadre** documentation website (https://jonesor.github.io/Rcompadre/) give further detailed coverage of the package's capabilities.

3 | Rage

The **Rage** package contains functions to facilitate the calculation of life-history metrics (Table 2) from MPMs. The guiding philosophy of the package centres on (a) augmenting the suite of life-history analyses that are implemented in R and (b) providing support for analyses—whether new in **Rage** or previously implemented elsewhere—to be conducted in a standardised way across large numbers of MPMs. Other functions are novel, such as estimates of the pace and shape of reproduction (Baudisch & Stott, 2019). Broadly, the functions fall into six categories (Figure 1b, Table 2):

1. Transformation: reshape, resize and reorder whole MPMs;
2. Life tables: convert MPMs to life tables and life table components;
3. Life-history traits: calculate life-history metrics;
4. Vital rates: extract and summarise the component vital rates of MPMs;
5. Visualisation: plot the life cycle graph;
6. Perturbation analyses: calculate sensitivity and (stochastic) elasticity of any demographic statistic to perturbations of MPM elements, vital rates or transition types.

To illustrate the functionality and inter-compatibility of functions among these categories, we describe a workflow that reconciles a common problem in comparative life-history analysis: the desired life-history metric requires an age-structured life table, but the available data are stage-structured MPMs. Although the mathematical descriptions for each step have long been available in the demographic literature, **Rage** both implements these as R functions and does so in a way that enables interoperability of function inputs and outputs. We provide in-depth vignettes for each group of functions at the **Rage** documentation website (https://jonesor.github.io/Rage/). However, several **Rage** functions, such as `mpm_to_table`, `entropy_...` and `shape_...`, rest on the production of age-based life tables from stage-based matrices and thus it is pertinent to outline this important aspect of **Rage** here.

To enable a broader range of life-history analyses on data from MPMs, **Rage** implements conversions of stage-structured MPMs to age-specific mortality and fertility life tables using methods developed by Cochran and Ellner (1992), Caswell (2001) and Caswell et al. (2018). These methods require that MPMs are decomposed into their constituent submatrices, **U**, and optionally **F** and/or **C** (see above) and the determination of the stage we consider to be the start of the life cycle (e.g. seed establishment, seed germination, etc.). In a nutshell, the method works by an iterative procedure whereby a synthetic cohort starting at age zero is projected using the matrix model. At every iteration, the cohort ages by one projection interval (often 1 year), and we can keep track of survivorship (l_x), the proportion of the original cohort that have survived each iteration. Fecundity is calculated in an analogous way. The result is a full life table that is readily available for use in analyses that require age-, rather than stage-structured trajectories of demographic processes. We direct readers to Caswell (2001), Caswell et al. (2018) and in the supplementary information of Jones et al. (2014).

Once an l_x trajectory is calculated, the other quantities of standard life tables can be calculated using standard life table calculations (Preston et al., 2000). In **Rage**, the function `mpm_to_table` applies these calculations to produce a life table that includes standard life table columns including age, survivorship, age-specific probability of death, force of mortality and remaining life expectancy. In addition, **Rage** provides functionality to calculate age trajectories for individual variables (i.e. subsets of the full life table) using the `mpm_to_...` set of functions (e.g. `mpm_to_lx`; Box 1).

Importantly, converting MPMs to life tables can introduce mathematical artefacts that compromise the resulting analyses. **Rage** provides functions to diagnose and, when possible, correct for these artefacts. All age-from-stage calculations produce age-trajectories that inevitably asymptote as a mathematical consequence of

BOX 2 Using Rage to calculate and visualise longevity

Here we demonstrate the use of **Rage**, focussing on the global analysis of mammalian longevity introduced in Box 1. We begin our mammal longevity analysis by adding columns to the data extracted from COMADRE (Box 1) that contain the two user-supplied arguments, matU and start_life, using the dplyr function mutate. We can then pair mutate with the base R function mapply to call the longevity function with each row's matU and start_life arguments and return the estimated longevity in a new column. Then we check the age of convergence to the quasi-stationary stage distribution (QSD), and filter the dataset so that it only includes matrices where the estimated longevity is less than or equal to the age at which QSD is reached.

```
 # Load package
library(Rage)
 # Add columns for matU and matF, then calculate generation time, longevity and
 # convergence
 # Filter to ensure that QSD is not reached before estimated longevity.
mammals <- mammals %>%
 mutate(
 matU = matU(.),
 start_life = mpm_first_active(.)
 ) %>%
 mutate(
 matF = matF(.),
 start_life = mpm_first_active(.)
 ) %>%
 mutate(gentime = mapply(gen_time, matU, matF)) %>%
 mutate(longevity = mapply(longevity, matU)) %>%
 mutate(convage = mapply(qsd_converge, matU)) %>%
 filter(longevity - convage <= 0)
library(khroma)
ggplot(mammals, aes(x = gentime, y = longevity)) +
 geom_point(aes(colour = Order)) +
 scale_color_manual(values = c(as.vector(colour("bright")(7)),"black")) +
 scale_x_continuous(trans = "log", breaks = c(2, 5, 10, 20, 40, 80)) +
 scale_y_continuous(trans = "log", breaks = c(2, 5, 10, 20, 40, 80, 160)) +
 labs(x = "Generation time (years)", y = "Longevity (years)") +
 geom_smooth(method = "lm", colour = "grey50") +
 theme_minimal()
 #> `geom_smooth()` using formula 'y ~ x'
```

As one might expect, there is a strong association between generation time and our measure of life span (Figure 3). It would of course be interesting to use more formal statistical methods to explore this (and similar relationships) further, for example to examine the variation in the scaling relationship across orders. When doing so, it will be important to carefully consider taxonomic and geographical or ecoregion bias in the dataset. In addition, researchers should carefully vet the included data for suitability—including a consideration of whether the models are based on pre- or post-reproduction censuses.

describing the vital rates as functions of discrete stages (Horvitz & Tuljapurkar, 2008). Regardless of how low the survival probabilities are in an MPM, there will be a non-zero probability that an individual could reach ages of 100, 10,000 or >1 million years. The exponential rate that these probabilities decay with increasing age is determined by the dominant eigenvalue of **U**, but even rapid decay can bias some life-history metrics (e.g. entropy and life span measures). **Rage** provides a convenient and principled way of correcting for this artefact by imposing a lower probability threshold defined by the degree of convergence to the quasi-stationary distribution (see also the supplementary information of Jones et al., 2014). In **Rage**, we do this by first scaling the right eigenvector (**w**) so that it sums to one and

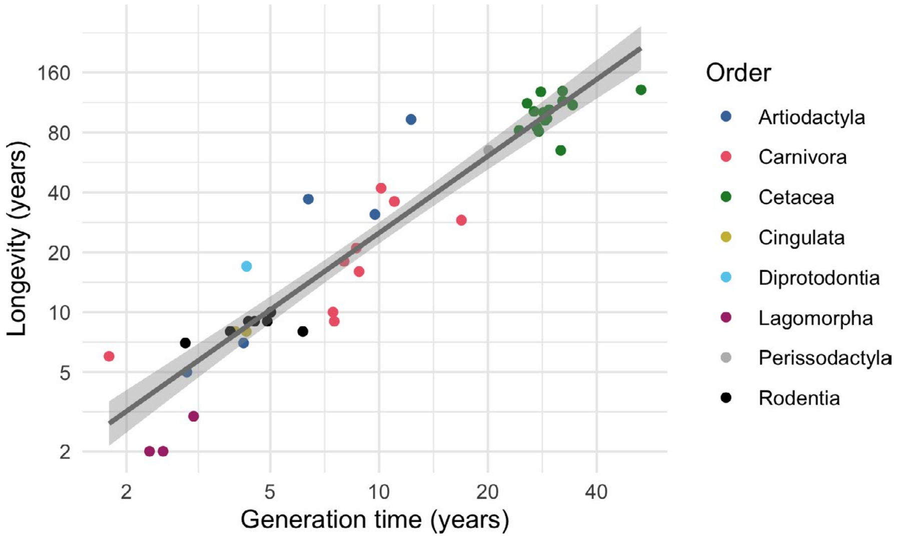

FIGURE 3 The relationship between estimated generation time and longevity (defined as the age that 1% of a synthetic cohort would reach, based on the MPM). The line represents the fit of an ordinary least-squared regression through the data. The slope is 1.28 (±0.07) and the intercept is 0.26 (±0.16); $R^2 = 0.90$; $F_{1,43} = 379$; $p < 0.001$

then, for each iteration of the age-from-stage calculations, we measure the convergence of the proportional cohort structure as $\Delta_x = 0.5 \, \|\mathbf{p}_x - \mathbf{w}\|$, where $\mathbf{p}_x$ is the proportional stage structure at the xth iteration of the age-from-stage calculations (i.e. at time *x*). When $\mathbf{p}_x$ eventually converges to equal **w**, Δ_x will equal 0. We can use this information to truncate the life tables produced from age-from-stage methods to, for example, ages where $\Delta_x > 0.05$. Furthermore, we may judge the reliability of age-from-stage methods by comparing the l_x trajectory with the Δ_x trajectory: If convergence is reached before l_x declines to, for example, 0.05 (i.e. 5% of the cohort remaining alive), we suggest reconsidering the use of this approach for that particular model.

In Box 2, we demonstrate the use of `Rage` via a global analysis of mammalian longevity introduced in Box 1. The life-history metric of interest is calculated with **Rage**'s `longevity` function—a novel implementation in this package—by projecting a hypothetical cohort of individuals with an MPM until only a user-defined (default: 1%) fraction of individuals from the initial cohort remain alive. Since only a single cohort is tracked, the function requires only the **U** submatrix (stage-specific survival and transition rates) as the demographic process input, which may be supplied directly by the user or extracted from a `CompadreDB` object using the `matU` function from `Rcompadre`.

The `longevity` function also requires us to define which stage we consider to be the start of the life cycle. This is fairly clear for most mammals but may be more subjective in some groups depending on the goals of the analysis (e.g. seed maturation versus germination for plants with a persistent seed bank). The **Rcompadre** function `mpm_first_active` facilitates scaling this task across a large number of MPMs by returning an integer index for the first active stage class (i.e. non-dormant), as defined by the original study author of the MPM. Like the results of `Rcompadre::cdb_flag`, we intend this to be used as a guide—not a replacement—for careful evaluation of suitability. It may be more appropriate to identify the start of life manually in some cases. Users may control the cohort survivorship threshold via the argument `lx_crit`. The default, 0.01 (=1%), may not be suitable for all organisms, and users may find that exploring other quantiles (e.g. 50%) offers a richer description of the age-at-death distribution. Finally, the function requires us to set a maximum age to consider (`xmax`, default = 1,000) as a pragmatic matter of computational speed. This default can be increased for exceptionally long-lived organisms, and we remind users that all measures of age in the **Rage** package use the projection interval of the MPM provided (see the `ProjectionInterval` metadata column for COM(P)ADRE data retrieved using `Rcompadre::cdb_fetch`). A final important caveat for the general use of **Rage** is that the life-history calculations, like most other MPM calculations, assume that the models are parameterised using post-reproductive census data. Therefore, outputs are likely to be incorrect if the models were parameterised using pre-reproductive data (see Kendall et al., 2019). We advise users to check the type of data included in analyses carefully and to exclude pre-reproductive matrices.

4 | CONCLUSIONS

The tools provided by **Rcompadre** and **Rage** facilitate efficient and at-scale use of an unrivalled database of demographic process rates and the calculation of numerous life-history and demographic metrics that are useful in ecology and evolution. In so doing, this pair of packages fills gaps and reduces overhead in the analytical workflow of comparative and macroecological demographic analyses. Although we designed the packages to operate together, **Rage** is also well-suited for general use with non-COM(P)ADRE matrix population models, whether in support of the analysis of new empirical MPMs or simulation-based theoretical studies of life history. We showcase the use of these packages to illustrate how they may be particularly useful in comparative demographic studies, for example, to address topics related to the evolution of life histories or comparative population dynamics across many species.

Users can obtain a complete index of the functions available in **Rcompadre** and **Rage** by running `?Rcompadre` and `?Rage`, respectively, in R, or by visiting the package documentation websites at https://jonesor.github.io/Rcompadre/ and https://jonesor.github.io/Rage/, respectively. Our ultimate hope is that democratising access to demographic data and analytic tools will empower a wide range of users to unlock the great potential of matrix population models. This will allow the community to further our basic understanding of life history, enable data-driven conservation management, and educate and inspire the next generation of population biologists.

ACKNOWLEDGEMENTS

We are grateful to the Max Planck Institute for Demographic Research for funding a workshop in winter 2017 to kickstart the development of these R packages. We thank the many attendees of our teaching workshops at numerous conferences, including the annual meetings of the British Ecological Society, Evolutionary Demography Society, International Convention for Conservation Biology and the Ecological Society of America. They inspired much of the functionality of these R packages. ORJ was supported by the Independent Research Fund Denmark (DFF-6108-00467). RS-G was supported by a NERC Independent Research Fellowship (NE/M018458/1). JC-C, ORJ, RS-G, CCT and CS were supported by an NSF Advances in Bioinformatics Development Award (#DBI-1661342). We thank D. Buss, J. Jones, J. Metcalf, H. Caswell and B. Kendall for contributing pieces of code and/or advice at early stages of this project. We are also grateful to Y. Vindenes and two anonymous reviewers for constructive comments on earlier drafts of our manuscript.

CONFLICT OF INTEREST

We declare no conflict of interest.

AUTHORS' CONTRIBUTIONS

O.R.J. and R.S.-G. conceived the packages; O.R.J., P.B., I.S., T.D.J., W.K.P., J.C.-C., S.L., G.R., C.C.T., C.S., P.C., J.J. and R.S.-G. wrote code and/or contributed to documentation; I.S. designed the logos and J.J. and P.C. created Figure 1; O.R.J. led the writing of the manuscript, and all authors contributed to the drafts and gave final approval to publication.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1111/2041-210X.13792.

DATA AVAILABILITY STATEMENT

Data used in the examples presented here are publicly available from www.compadre-db.org

ORCID

Owen R. Jones https://orcid.org/0000-0001-5720-4686
Patrick Barks https://orcid.org/0000-0002-5947-8151
Iain Stott https://orcid.org/0000-0002-1399-3596
Tamora D. James https://orcid.org/0000-0003-1363-4742
Sam Levin https://orcid.org/0000-0002-3289-9925
John Jackson https://orcid.org/0000-0002-4563-2840
Roberto Salguero-Gómez https://orcid.org/0000-0002-6085-4433

REFERENCES

Baudisch, A. (2011). The pace and shape of ageing. *Methods in Ecology and Evolution*, *2*(4), 375–382. https://doi.org/10.1111/j.2041-210X.2010.00087.x

Baudisch, A., & Stott, I. (2019). A pace and shape perspective on fertility. *Methods in Ecology and Evolution*, *10*(11), 1941–1951. https://doi.org/10.1111/2041-210X.13289

Capdevila, P., Beger, M., Blomberg, S. P., Hereu, B., Linares, C., & Salguero-Gómez, R. (2020). Longevity, body dimension and reproductive mode drive differences in aquatic versus terrestrial life-history strategies. *Functional Ecology*, *34*(8), 1613–1625. https://doi.org/10.1111/1365-2435.13604

Caswell, H. (2001). *Matrix population models*. Sinauer Associates Incorporated.

Caswell, H., de Vries, C., Hartemink, N., Roth, G., & van Daalen, S. F. (2018). Age × stage-classified demographic analysis: A comprehensive approach. *Ecological Monographs*, *88*(4), 560–584.

Cochran, M. E., & Ellner, S. (1992). Simple methods for calculating age-based life history parameters for stage-structured populations. *Ecological Monographs*, *62*(3), 345–364. https://doi.org/10.2307/2937115

Coutts, S. R., Salguero-Gómez, R., Csergő, A. M., & Buckley, Y. M. (2016). Extrapolating demography with climate, proximity and phylogeny: Approach with caution. *Ecology Letters*, *19*(12), 1429–1438. https://doi.org/10.1111/ele.12691

de Kroon, H., Plaisier, A., van Groenendael, J., & Caswell, H. (1986). Elasticity: The relative contribution of demographic parameters to population growth rate. *Ecology*, *67*(5), 1427–1431. https://doi.org/10.2307/1938700

de Kroon, H., van Groenendael, J., & Ehrlén, J. (2000). Elasticities: A review of methods and model limitations. *Ecology*, *81*(3), 607–618. https://doi.org/10.1890/0012-9658(2000)081[0607:EAROMA]2.0.CO;2

de la Cruz Rot, M. (2019). *Rramas: Matrix population models*. https://CRAN.R-project.org/package=Rramas

Demetrius, L. (1978). Adaptive value, entropy and survivorship curves. *Nature*, *275*(5677), 213–214.

Ehrlén, J. (2000). The dynamics of plant populations: Does the history of individuals matter? *Ecology*, *81*(6), 1675–1684. https://doi.org/10.1890/0012-9658(2000)081[1675:TDOPPD]2.0.CO;2

Franco, M., & Silvertown, J. (2004). A comparative demography of plants based on elasticities of vital rates. *Ecology*, *85*(2), 531–538. https://doi.org/10.1890/02-0651

Healy, K., Ezard, T. H. G., Jones, O. R., Salguero-Gómez, R., & Buckley, Y. M. (2019). Animal life history is shaped by the pace of life and the distribution of age-specific mortality and reproduction. *Nature Ecology & Evolution*, *3*(8), 1217–1224. https://doi.org/10.1038/s41559-019-0938-7

Horvitz, C. C., & Tuljapurkar, S. (2008). Stage dynamics, period survival, and mortality plateaus. *The American Naturalist*, *172*(2), 203–215. https://doi.org/10.1086/589453

James, T. D., Salguero-Gómez, R., Jones, O. R., Childs, D. Z., & Beckerman, A. P. (2020). Bridging gaps in demographic analysis with phylogenetic imputation. *Conservation Biology*, *35*(4), 1210–1221. https://doi.org/10.1111/cobi.13658

Jones, O. R., Ezard, T. H. G., Dooley, C., Healy, K., Hodgson, D. J., Mueller, M., Townley, S., & Salguero-Gómez, R. (2020). My family and other animals: Human demography under a comparative cross-species lens. In O. Burger, R. Lee, & R. Sear (Eds.), *Human evolutionary demography* (pp. 1–31). OSF. https://osf.io/ehp92/

Jones, O. R., Scheuerlein, A., Salguero-Gómez, R., Camarda, C. G., Schaible, R., Casper, B. B., Dahlgren, J. P., Ehrlén, J., García, M. B., Menges, E. S., Quintana-Ascencio, P. F., Caswell, H., Baudisch, A., & Vaupel, J. W. (2014). Diversity of ageing across the tree of life. *Nature*, *505*(7482), 169–173.

Jouvet, L., Rodríguez-Rojas, A., & Steiner, U. K. (2018). Demographic variability and heterogeneity among individuals within and among clonal bacteria strains. *Oikos*, *127*(5), 728–737. https://doi.org/10.1111/oik.04292

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.

Keyfitz, N., & Caswell, H. (2005). *Applied mathematical demography*. Springer Science & Business Media.

Leslie, P. H. (1945). On the use of matrices in certain population mathematics. *Biometrika*, *33*, 183–212. https://doi.org/10.1093/biomet/33.3.183

Leslie, P. H. (1948). Some further notes on the use of matrices in population mathematics. *Biometrika*, *35*(3/4), 213–245. https://doi.org/10.1093/biomet/35.3-4.213

Matlab. (2010). *version 7.10.0 (R2010a)*. The MathWorks Inc.

Morris, W. F., Altmann, J., Brockman, D. K., Cords, M., Fedigan, L. M., Pusey, A. E., Stoinski, T. S., Bronikowski, A. M., Alberts, S. C., & Strier, K. B. (2011). Low demographic variability in wild primate populations: Fitness impacts of variation, covariation, and serial correlation in vital rates. *The American Naturalist*, *177*(1), E14–E28. https://doi.org/10.1086/657443

Morris, W. F., & Doak, D. F. (2002). *Quantitative conservation biology*. Sinauer.

Okland, R. H. (1995). Population biology of the clonal moss *Hylocomium splendens* in Norwegian boreal spruce forests. I. Demography. *The Journal of Ecology*, *83*(4), 697. https://doi.org/10.2307/2261637

Preston, S., Heuveline, P., & Guillot, M. (2000). *Demography: Measuring and modeling population processes*. Wiley.

R Core Team. (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Salguero-Gómez, R., Jones, O. R., Archer, C. R., Bein, C., Buhr, H., Farack, C., Gottschalk, F., Hartmann, A., Henning, A., Hoppe, G., Römer, G., Ruoff, T., Sommer, V., Wille, J., Voigt, J., Zeh, S., Vieregg, D., Buckley, Y. M., Che-Castaldo, J., … Vaupel, J. W. (2016). COMADRE: A global data base of animal demography. *Journal of Animal Ecology*, *85*(2), 371–384. https://doi.org/10.1111/1365-2656.12482

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., Buhr, H., Farack, C., Gottschalk, F., Hartmann, A., Henning, A., Hoppe, G., Römer, G., Runge, J., Ruoff, T., … Vaupel, J. W. (2015). The compadre Plant Matrix Database: An open online repository for plant demography. *Journal of Ecology*, *103*(1), 202–218. https://doi.org/10.1111/1365-2745.12334

Salguero-Gómez, R., & Plotkin, J. B. (2010). Matrix dimensions bias demographic inferences: Implications for comparative plant demography. *American Naturalist*, *176*(6), 710–722. https://doi.org/10.1086/657044

Shefferson, R. P., Kurokawa, S., & Ehrlén, J. (2021). lefko3: Analysing individual history through size-classified matrix population models. *Methods in Ecology and Evolution*, *12*(2), 378–382. https://doi.org/10.1111/2041-210x.13526

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*(5), 797–802. https://doi.org/10.1111/j.2041-210x.2012.00222.x

Stott, I., Townley, S., Carslake, D., & Hodgson, D. J. (2010). On reducibility and ergodicity of population projection matrix models. *Methods in Ecology and Evolution*, *1*(3), 242–252. https://doi.org/10.1111/j.2041-210X.2010.00032.x

Stubben, C., & Milligan, B. (2007). Estimating and analyzing demographic models using the popbio package in R. *Journal of Statistical Software*, *22*(11). https://doi.org/10.18637/jss.v022.i11

Takada, T., & Kawai, Y. (2020). An analysis of elasticity vector distribution specific to semelparous species using randomly generated population projection matrices and the COMPADRE Plant Matrix Database. *Ecological Modelling*, *431*, 109125. https://doi.org/10.1016/j.ecolmodel.2020.109125

Wickham, H. (2019). *Advanced R* (2nd ed). CRC Press.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T., Miller, E., Bache, S., Müller, K., Ooms, J., Robinson, D., Seidel, D., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., & Grolemund, G. (2016). *R for data science: Import, tidy, transform, visualize, and model data*. O'Reilly Media Inc.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Jones, O. R., Barks, P., Stott, I., James, T. D., Levin, S., Petry, W. K., Capdevila, P., Che-Castaldo, J., Jackson, J., Römer, G., Schuette, C., Thomas, C. C., & Salguero-Gómez, R. (2022). **Rcompadre** and **Rage**—Two R packages to facilitate the use of the COMPADRE and COMADRE databases and calculation of life-history traits from matrix population models. *Methods in Ecology and Evolution*, 00, 1–12. https://doi.org/10.1111/2041-210X.13792