`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, Campusvej 55, 5230 Odense C, Denmark.

[2] School of Life Sciences, University of Lincoln, Brayford Pool, Lincoln, LN6 7TS, UK.

[3] Department of Animal and Plant Sciences, University of Sheffield, Western Bank, Sheffield S10 2TN, UK.

[4] Helmholtz-Centre for Environmental Research – UFZ, Martin Luther University Halle-Wittenberg, Puschstrasse 4, 04103 Leipzig, Germany.

[5] Department of Plant and Microbial Biology, North Carolina State University, Raleigh, North Carolina 27607, USA.

[6] School of Biological Sciences, University of Bristol, 24 Tyndall Ave, BS8 1TQ, Bristol, UK.

[7] Alexander Center for Applied Population Biology, Conservation & Science Department, Lincoln Park Zoo, Chicago, Illinois 60614, USA.

[8] Department of Zoology, University of Oxford, 11a Mansfield Rd, Oxford, OX1 3SZ, UK.

[9] Max Planck Institute for Demographic Research, 1 Konrad-Zuße Straße, Rostock, DE 18057, Germany.

[*] Corresponding author: jones@biology.sdu.dk

Summary

1. Matrix population models (MPMs) are an important tool in the arsenal of biologists seeking to understand the causes and consequences of variation in vital rates (e.g. survival, reproduction) across life cycles. MPMs describe the age- or stage-structured demography of organisms and represent the life history of a population during a particular time frame at a specific geographic location.
2. The COMPADRE Plant Matrix Database and COMADRE Animal Matrix Database are the most extensive resources for MPM data, collectively containing >12,000 MPMs 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 users by requiring advanced knowledge to handle diverse data structures and program custom analysis functions.
3. To close this knowledge gap, we present two R packages designed to (i) facilitate the use of these databases by providing functions to acquire, check, and manage the MPM data contained in COMPADRE and COMADRE (**Rcompadre**), and (ii) expand the range of life history traits that can be calculated from MPMs in support of ecological and evolutionary analyses (**Rage)**. We provide vignettes to illustrate the use of both **Rcompadre** and **Rage**.
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.

Keywords: Ageing, Age-structure population model, Life history strategy, Matrix population model, Population projection model, Population dynamics, Stage-structured population model

Introduction

Matrix population models (MPMs, hereafter) have become a commonplace tool in the arsenal of 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 (st)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, Plaisier, van Groenendael, & Caswell, 1986; de Kroon, van Groenendael, & Ehrlén, 2000). These outputs 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) and in myriad ecoregions.

Since the introduction of MPMs in the 1940s (Leslie 1945, 1948), researchers have published thousands of MPMs for over a thousand 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 sister databases now contain more than 12,000 MPMs for more than 1,100 species (COMPADRE: 8,870 matrices for 759 species; COMADRE: 3,366 matrices for 416 species, at the time of writing in February 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).

COMPADRE and COMADRE store and provide MPMs and their associated metadata in a hierarchical structure that can be both a barrier to use and an entry point for errors. Individual MPMs are two-dimensional square matrices, and the size of the matrices can vary widely across species and studies. Moreover, most MPMs (**A**) in the databases are partitioned into their three constituent process-based submatrices such that **A** = **U** + **F** + **C**. Here, submatrix **U** contains transitions related to survival and growth/development, submatrix **F** contains sexual reproduction, and submatrix **C** contains clonal reproduction. Thus, in most cases, each matrix is represented by these four matrices (**A**, **U**, **F** and **C**) alongside information about the life cycle stages used in the MPM. Each MPM entry is also associated with over 40 metadata variables (e.g. citation to original works, taxonomy, geography, etc., detailed in Salguero-Gómez et al., 2015 & 2016). This nested structure means that the dataset cannot be imported by ordinary spreadsheet software, such as Excel, that can only accommodate 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 ecosystems*"). 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 MPMs. For example, **`popdemo`** (Stott, Hodgson, & Townley, 2012) focuses on the calculation of metrics related to transient population dynamics and transfer function analyses; **`popbio`** (Stubben, Milligan, & Others, 2007) provides functions to accomplish many (but not all) of the analyses found in the textbooks of Caswell (2001) and Morris & 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, Kurokawa, & Ehrlén, 2021) provides tools that allow the inclusion of information on individual histories, which could influence population dynamics, into MPM analyses (see Ehrlén, 2000). However, these existing packages provide few tools for life history analysis that COMPADRE and COMADRE are intended to enable. They also 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. Researchers that wanted to make calculations related to, for example, longevity, senescence, or age at maturity, have needed to write their own code based on published equations in mathematics-heavy work, which has been a barrier to the broader adoption of these methods.

Here, we introduce two R packages that aim to close these knowledge and usability gaps and which readily combine to support MPM analysis workflows. The first package, **Rcompadre**, is designed to facilitate acquiring, checking, and managing the rich, hierarchical MPM data in COMPADRE and COMADRE. For example, this library includes tools to filter (subset) the databases based on metadata archived in these resources (e.g. by biome, by taxonomic group). In addition to base R style syntax for these tasks, **Rcompadre** integrates **tidyverse** (Wickham et al., 2019) functionality to improve usability. The second package, **Rage**, facilitates the calculation of life history traits from arbitrarily large numbers of MPMs supporting comparative analyses. These include important variables such as life tables, mean life expectancy, generation time, among several others. We illustrate the use of these two R packages sequentially. We first showcase downloading, subsetting, and preparing MPM data for analysis over a sizeable taxonomic breadth using **Rcompadre**. Second, we highlight some key applications of **Rage** to calculate ecologically and evolutionarily relevant metrics to test hypotheses related to life history theory. Third, we demonstrate the functional integration of **Rcompadre** and **Rage** and how investigators can use them in tandem to tackle pertinent questions in ecology and evolution. These analyses exemplify the use of two synergistic R packages that researchers can harness for future work in ecology, evolution, and conservation biology. We illustrate typical use cases of the two packages in Fig. 1.

Rcompadre

Rcompadre contains functions to facilitate downloading and using MPMs and ancillary metadata data from the COMPADRE and COMADRE databases (Fig. 1a). This library establishes and defines a new class of R data object, `CompadreDB`, which we have designed to improve the ease with which users can manipulate the MPM data provided by the COM(P)ADRE databases. Briefly, objects of class `CompadreDB` are S4 objects[1] that include two slots: the 'data' slot, which contains a `tibble`-style data frame (Wickham & Grolemund, 2016) with a list-column of MPMs and columns of metadata, and the 'version' slot which contains version information, including version number, date created, and a link to the database user agreement. This information will improve the reproducibility of analyses. In addition, we have created the `CompadreMat` class, which formally defines how MPMs are represented in a `CompadreDB` object. Establishing these explicit classes has allowed us to define how users may access the data contained in the objects. For example, users can access and replace columns of data using the standard `x$name` and `x$name <- value` methods. In addition, we provide the functionality to access the matrices directly, for example, using the functions `matA` or `matU` to access all **A** matrices or **U** matrices 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 analysis (for example, see recent studies by Coutts et al. (2016), Takada & Kawai (2020), James et al. (2020), Healy et al. (2019), Capdevila et al. (2020) and Jones et al. (2020)). The class definition has also allowed us to create our own `CompadreDB`-specific methods for existing R functionality. Perhaps the most useful of these are the **`tidyverse`** methods (Wickham et al., 2019), familiar to many R users. In short, users can use the majority of the existing functions from the **`tidyverse`** package **`dplyr`** to manipulate COM(P)ADRE data. There are methods for the `filter`, `arrange`, `mutate`, `select`, `summarise`, `rename` and `join` functions. The provision of these **`tidyverse`** methods also means that **`Rcompadre`** benefits from the piping (e.g. `%>%`) functionality of **`magrittr`**, improving script fluency and readability. Examples of this functionality can be found in the vignettes at the package development pages.

[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 both within and across , 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.

The main functions of **`Rcompadre`** are prefixed with `cdb_` , which stands for "compadre database", followed by a meaningful verb (e.g. `cdb_fetch`, or `cdb_compare`). Table 1 summarises the most important **`Rcompadre`** functions.

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.

The first task is to install the latest version of **`Rcompadre`**. **`Rcompadre`** is developed on the GitHub platform and can be installed using `remotes::install_github`.

```r
install.packages("remotes")
remotes::install_github("jonesor/Rcompadre")
```

After ensuring that the package is installed we load it, along with other useful packages.

```r
library(Rcompadre)
library(tidyverse)
```

After that, we can download the latest version of the COMPADRE or COMADRE databases from the website using `cdb_fetch`. We could also load local files with the same function (`comadre <- cdb_fetch("PATH/TO/FILE/COMADRE_v.4.21.1.0.RData")`). After loading the data, a message is displayed stating the database version number, release date, and links to the user agreement and citation information.

```r
comadre <- cdb_fetch("comadre")
#> This is COMADRE version 4.21.1.0 (release date Jan_25_2021)
#> See user agreement at https://compadre-db.org/Help/UserAgreement
#> See how to cite at https://compadre-db.org/Help/HowToCite
```

Data management

The COMADRE database includes metadata associated with each matrix including taxonomic information, geolocation, and details of the source publication (see the User Guide at www.compadre-db.org or Salguero-Gomez et al. 2015, 2016 for full metadata documentation). We can get an idea of the richness of the metadata with the `names` function.

```
names(comadre)
#>  [1] "mat"                "MatrixID"
#>  [3] "SpeciesAuthor"      "SpeciesAccepted"
#>  [5] "CommonName"         "Kingdom"
#>  [7] "Phylum"             "Class"
#>  [9] "Order"              "Family"
#> [11] "Genus"              "Species"
#> [13] "Infraspecies"       "InfraspeciesType"
#> [15] "OrganismType"       "DicotMonoc"
#> [17] "AngioGymno"         "Authors"
#> [19] "Journal"            "SourceType"
#> [21] "OtherType"          "YearPublication"
#> [23] "DOI_ISBN"           "AdditionalSource"
#> [25] "StudyDuration"      "StudyStart"
#> [27] "StudyEnd"           "ProjectionInterval"
#> [29] "MatrixCriteriaSize" "MatrixCriteriaOntogeny"
#> [31] "MatrixCriteriaAge"  "MatrixPopulation"
#> [33] "NumberPopulations"  "Lat"
#> [35] "Lon"                "Altitude"
#> [37] "Country"            "Continent"
#> [39] "Ecoregion"          "StudiedSex"
#> [41] "MatrixComposite"    "MatrixSeasonal"
#> [43] "MatrixTreatment"    "MatrixCaptivity"
#> [45] "MatrixStartYear"    "MatrixStartSeason"
#> [47] "MatrixStartMonth"   "MatrixEndYear"
#> [49] "MatrixEndSeason"     "MatrixEndMonth"
#> [51] "CensusType"          "MatrixSplit"
#> [53] "MatrixFec"           "Observations"
#> [55] "MatrixDimension"     "SurvivalIssue"
#> [57] "_Database"           "_PopulationStatus"
#> [59] "_PublicationStatus"
```

The MPMs 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.

We first explore taxonomic groups available in the dataset.

```
table(comadre$Class)
#> 
#>     Actinopterygii        Adenophorea           Amphibia
#>                196                  4                 24
#>           Anthozoa          Arachnida         Ascidiacea
#>                102                  5                 18
#>               Aves           Bivalvia       Branchiopoda
#>                409                 30                  7
#> Cephalaspidomorphi         Clitellata       Demospongiae
#>                  9                  3                 13
#>          Diplopoda         Echinoidea     Elasmobranchii
#>                  1                 11                  4
#>         Gastropoda       Gymnolaemata            Insecta
#>                 20                 10                 73
#>       Malacostraca           Mammalia        Maxillopoda
#>                 30               2181                 11
#>         Polychaeta           Reptilia    Revtraviricetes
#>                 19                168                  2
#>           Scyphozoa          Secernentea       Spirochaetes
#>                   2                    4                  9
#>           Thaliacea
#>                   1
```

We see that COMADRE contains information for many groups (the number of matrices available is tallied for each taxonomic Class). In this case, we are only interested in mammals. **Rcompadre** allows us to subset the database easily using the function filter from `dplyr` as if our `CompadreDB` object was an ordinary data frame.

```
mammals <- comadre %>%
  filter(Class == "Mammalia")

table(mammals$Order)
#>
#>    Artiodactyla     Carnivora        Cetacea
#>             137           223             58
#>      Chiroptera     Cingulata Dasyuromorphia
#>               1             3              3
#> Didelphimorphia Diprotodontia     Lagomorpha
#>               1            28             10
#>  Perissodactyla      Primates    Proboscidea
#>               1          1557              3
#>        Rodentia       Sirenia
#>             153             3
```

Data checking

Although COMPADRE and COMADRE data undergo rigorous quality checks before being made available, not all data will meet the inclusion requirements for a particular analysis. **`Rcompadre`** includes several general functions for checking the data, though we strongly encourage investigators to perform additional checks that may be necessary to determine the suitability of a record for inclusion.

For example, a minority of studies published only the projection matrix, **A**, thereby preventing it from being decomposed into the **U**, **F** and **C** submatrices. Because some life history traits require one or more of these submatrices, investigators may wish to find and filter out records that cannot be used. In our example analysis, calculating longevity requires a **U** submatrix with no missing values. We use the function `cdb_flag` to identify MPMs that do not meet these criteria by specifying `checks = c("check_NA_U","check_zero_U")` (see `?cdb_flag` for all available data property checks). Each check adds a logical column to the provided `CompadreDB` object. Here two columns are added indicating whether each record contains missing **U** matrix elements (`check_NA_U`) and whether these **U** matrices are all zeros (`check_zero_U`).

```
mammals <- cdb_flag(mammals, checks = c("check_NA_U", "check_zero_U"))
```

We can summarise the outcome of each check using `table`.

```
table(mammals$check_NA_U)
#>
#> FALSE  TRUE
#>  2168     13
table(mammals$check_zero_U)
```

```
#> 
#> FALSE  TRUE 
#>  2172     9
```

Then we can `filter` our data to include only data with suitable **U** matrices, indicated by `FALSE` values in all columns added by `cdb_flag`.

```
mammals <- mammals %>%
  filter(check_NA_U == FALSE, check_zero_U == FALSE)
```

Finally, most COM(P)ADRE records are geolocated (`Lat` and `Lon` columns) to enable spatially-explicit analyses. These metadata make it easy to visualise the spatial extent of our data set using tools in the **`ggplot2`** and **`maps`** packages (Fig. 2A).

```
globalDist <- ggplot(mammals, aes(x = Lon, y = Lat)) +
  borders(database = "world", fill = "grey80", col = NA) +
  geom_point(alpha = 0.9) +
  labs(x = NULL, y = NULL) +
  theme_minimal()

globalDist
```

With the subsetted and cleaned data in hand, we are ready to continue the workflow in the **`Rage`** package to calculate mammalian longevity.

Rage

`Rage` contains functions to facilitate the calculation of life history metrics (Table 2) from MPMs. Some of these functions are “standard” matrix calculations (e.g. mean and variance of life expectancy) available in the other R packages but are nevertheless implemented in **`Rage`** to support work with large MPM datasets. Other functions are novel, such as estimates of the pace and shape of reproduction (Baudisch & Stott, 2019). Broadly, the functions can be split into groups based on their application (Fig 1B, Table 2): the calculation of life history traits, the production of life tables, the calculation of vital rates, perturbation analysis, MPM transformation, and visualisation. Here we outline the most important of these functions, and we provide separate vignettes that explain the functionality of each group of functions in depth.

In the following section, we demonstrate the use of **`Rage`**, focussing on the global analysis of mammalian longevity introduced above. We begin by installing the package from GitHub.

```
install.packages("remotes")
remotes::install_github("jonesor/Rage")
```

Then load it to make **`Rage`** functions available in the working environment.

```
library(Rage)
```

Among these is the `longevity` function, which simulates the ages at death of a synthetic cohort of individuals following the demographic pattern described by the MPM and returns the expected longevity. The function documentation (`?longevity`) tells us that it requires the **U** matrix, which may be supplied directly by the user or extracted from a `CompadreDB` object using the `matU` function from **`Rcompadre`**.

The function also requires us to define which stage we consider to be the beginning of life. 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 *vs* germination for plants with a persistent seed bank). The **`Rcompadre`** function `mpm_first_active` automates the task 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. Here too, we encourage investigators to evaluate the suitability of this automated function during analysis carefully; it may be better to identify the start of life manually in some cases.

The longevity function also accepts the argument `lx_crit`, which defines the critical threshold of survivorship that defines longevity. The default, 0.01, corresponds to the age at which only 1% of the synthetic cohort remain alive. Still, users may find other quantiles (e.g., 50%) offer additional valuable descriptions of the age at death distribution. Finally, the function requires us to set a maximum age to consider (`xmax`, which defaults to 1000 for computational speed but can be increased for exceptionally long-lived organisms).

We begin our mammal longevity analysis by adding columns to the data extracted from COMADRE that contain the two user-supplied arguments, `matU` and `start_life`, using the **`tidyverse`** verb `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.

```
mammals <- mammals %>%
  mutate(
      matU = matU(.),
      start_life = mpm_first_active(.)
  ) %>%
  mutate(longevity = mapply(longevity, matU, start_life))
```

With only a few dozen lines of code, we have produced a large dataset of mammalian longevity that we can use to explore patterns of variation or test hypotheses about its drivers. Here, our goals are exploratory, and we'll construct plots to visualise patterns of longevity against covariates in the COMPADRE and COMADRE metadata. First, we will map the locations of our mammal MPM records and colour them by longevity (Fig. 2).

```
globalDist_longevity <- ggplot(mammals, aes(Lon, Lat)) +
  borders(database = "world", fill = "grey80", col = NA) +
  geom_point(aes(colour = log10(longevity)), alpha = 0.4, size = 0.75) +
  scale_colour_viridis_c(option = "C",name = expression(log[10]*"(longevity,
yrs)")) +
  labs(x = "longitude", y = "latitude") +
  ylim(-90,90) + xlim(-180,180)+
  theme_minimal()
```

```
globalDist_longevity
```

Because geographic overrepresentation in Europe and North America makes patterns difficult to discern from the map alone, we can follow up by plotting longevity against the latitude and ecoregion values extracted from the metadata. A visual inspection of the data shows an apparent decline in longevity at intermediate latitudes (Fig. 3A), and variation among ecoregions (Fig. 3B).

```
longevity_lat <- ggplot(mammals, aes(x = abs(Lat), y = log10(longevity))) +
  geom_point(alpha = 0.4) +
  labs(x = "Absolute latitude", y = expression(log[10]*"(longevity, yrs)")) +
  theme_minimal() +
  geom_smooth()

longevity_lat

longevity_ecoregion <- ggplot(mammals, aes(
  x = Ecoregion, y = log10(longevity),
  fill = Ecoregion
)) +
  geom_boxplot(show.legend = FALSE) +
  geom_jitter(alpha = 0.5, width = 0.1, show.legend = FALSE) +
  labs(x = "Ecoregion", y = expression(log[10]*"(longevity, yrs)")) +
  coord_flip() +
  theme_minimal()

longevity_ecoregion
```

These patterns could well be driven by data biases but would make an interesting topic for future exploration with more formal statistical methods. With the broad range of tools at our fingertips in **Rcompadre** and **Rage**, we can thus begin to address such detailed comparative questions with increased ease.

Conclusions

The tools provided by **Rcompadre** and **Rage** facilitate efficient use of an unrivalled database of demographic data 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 a fundamental gap in the analytical workflow of comparative and macroecological demographic analysis. 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 demographers.

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. 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.

Authors contributions

ORJ and RS-G conceived the packages. ORJ, PB, IS, TJ, WKP, JC-C, SL, GR, CCT, CS, PC, JJ and RS-G wrote code and/or contributed to documentation. IS designed the logos and JJ and PC created Fig. 1. ORJ led the writing of the manuscript, and all authors contributed to the drafts and gave final approval to publication.

Data availability

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

Supplementary materials

We provide several vignettes which guide users through most of the functionality of **`Rcompadre`** and **`Rage`**. These vignettes are available at the package development web pages at https://jonesor.github.io/Rcompadre/ and https://jonesor.github.io/Rage/, under "Articles" in the dropdown menu.

`Rcompadre:`

1. Getting started with Rcompadre
2. Using Rcompadre with the tidyverse
3. Vectorising with Rcompadre
4. Obtaining references

`Rage:`

1. Getting started with Rage
2. Deriving vital rates from an MPM
3. Deriving life history traits from an MPM
4. Age-from-stage analyses
5. Ternary plots

References

Baudisch, A. (2011). The pace and shape of ageing. *Methods in Ecology and Evolution*, *2*(4), 375–382.

Baudisch, A., & Stott, I. (2019). A pace and shape perspective on fertility. *Methods in Ecology and Evolution*, *10*(11), 1941–1951.

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.

Caswell, H. (2001). *Matrix Population Models*. Sinauer Associates Incorporated.

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.

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.

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.

de Kroon, H., van Groenendael, J., & Ehrlén, J. (2000). Elasticities: A review of methods and model limitations. *Ecology*, *81*(3), 607–618.

de la Cruz Rot, M. (2019). Rramas: Matrix Population Models. Retrieved from 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.

Franco, M., & Silvertown, J. (2004). A comparative demography of plants based on elasticities of vital rates. *Ecology*, *85*(2), 531–538.

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.

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: The Journal of the Society for Conservation Biology*. doi:10.1111/cobi.13658

Jones, O. R., Ezard, T. H. G., Dooley, C., Healy, K., Hodgson, D. J., Mueller, M., … 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.

Jones, O. R., Scheuerlein, A., Salguero-Gómez, R., Camarda, C. G., Schaible, R., Casper, B. B., … Vaupel, J. W. (2014). Diversity of ageing across the tree of life. *Nature*, *505*(7482), 169–173.

Keyfitz, N., & Caswell, H. (2005). *Applied Mathematical Demography*. New York: Springer Science & Business Media.

Leslie, P. H. (1945). On the use of matrices in certain population mathematics. *Biometrika*, *33*, 183–212.

Leslie, P. H. (1948). Some Further Notes on the Use of Matrices in Population Mathematics. *Biometrika*, *35*(3/4), 213–245.

Matlab. (2010). *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc.

Morris, W. F., Altmann, J., Brockman, D. K., Cords, M., Fedigan, L. M., Pusey, A. E., … Strier, K. B. (2011). Low Demographic Variability in Wild Primate Populations: Fitness Impacts of Variation, Covariation, and Serial Correlation in Vital Rates. *American Naturalist*, *177*(1), E14–E28.

Morris, W. F., & Doak, D. F. (2002). *Quantitative Conservation Biology*. Sunderland MA, USA: Sinauer.

Okland, R. H. (1995). Population biology of the clonal moss *Hylocomium splendens* in Norwegian boreal spruce forests. I. Demography. *Journal of Ecology*, *83*(4), 697–712.

R Core Team. (2021). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Salguero-Gómez, R., Jones, O. R., Archer, C. R., Bein, C., de Buhr, H., Farack, C., … Vaupel, J. W. (2016). COMADRE: a global database of animal demography. *Journal of Animal Ecology*, *85*(2), 371–384.

Salguero-Gómez, R., Jones, O. R., Archer, C. R., Buckley, Y. M., Che-Castaldo, J., Caswell, H., … Vaupel, J. W. (2015). The COMPADRE Plant Matrix Database: an open online repository for plant demography. *Journal of Ecology*, *103*(1), 202–218.

Salguero-Gómez, R., & Plotkin, J. B. (2010). Matrix dimensions bias demographic inferences: implications for comparative plant demography. *American Naturalist*, *176*(6), 710–722.

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

Stott, I., Hodgson, D. J., & Townley, S. (2012). popdemo : an R package for population demography using projection matrix analysis: popdemo : an R package. *Methods in Ecology and Evolution / British Ecological Society*, *3*(5), 797–802.

Stubben, C., Milligan, B., & Others. (2007). Estimating and analyzing demographic models using the popbio package in R. *Journal of Statistical Software*, *22*(11), 1–23.

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.

Wickham, H. (2019). *Advanced R, Second Edition*. CRC Press.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686.

Wickham, H., & Grolemund, G. (2016). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. Sebastopol, CA, USA: O'Reilly Media, Inc.

Figure 1

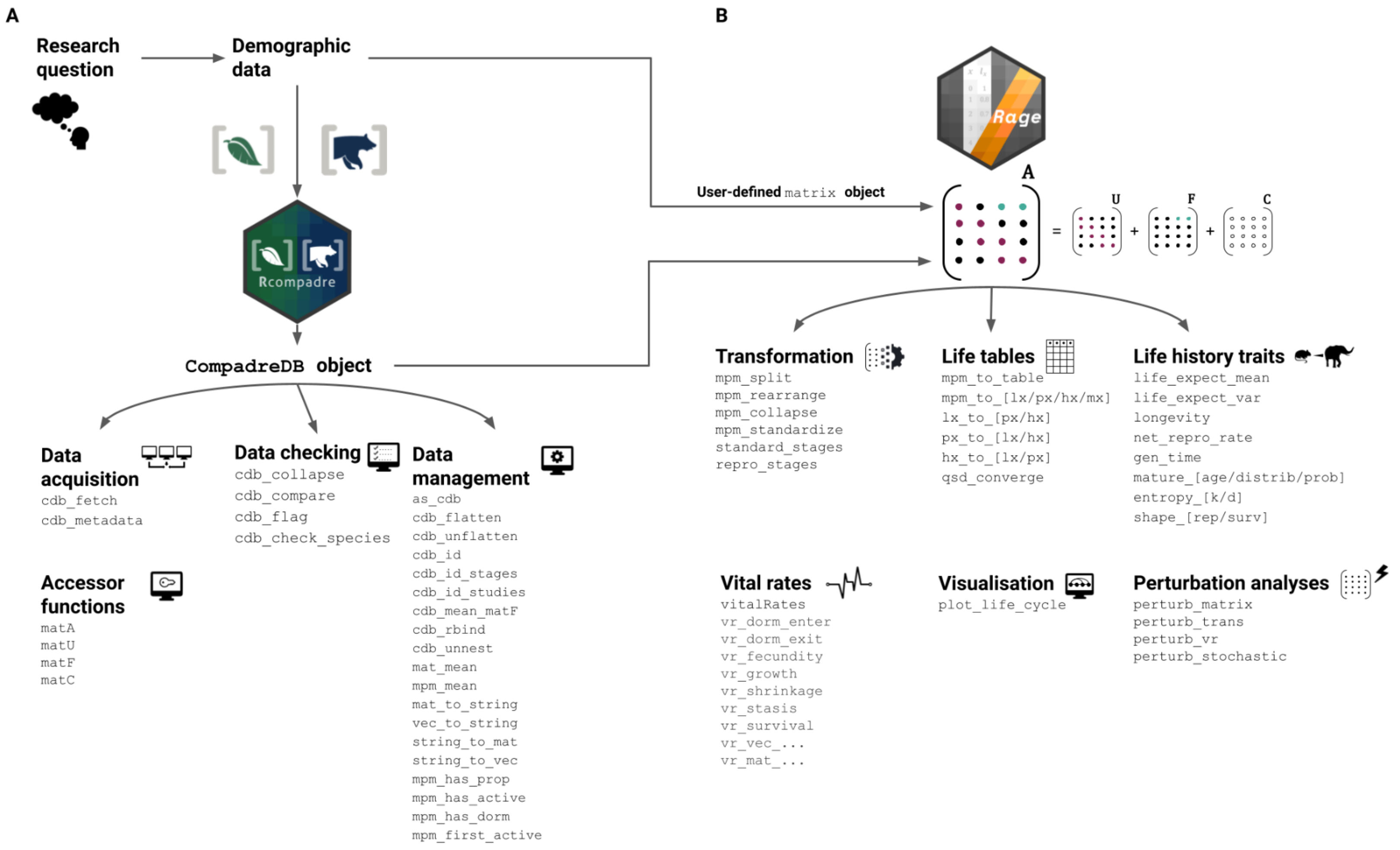
A
Research question
Demographic data
Rcompadre
CompadreDB object
Data acquisition
cdb_fetch
cdb_metadata
Data checking
cdb_collapse
cdb_compare
cdb_flag
cdb_check_species
Data management
as_cdb
cdb_flatten
cdb_unflatten
cdb_id
cdb_id_stages
cdb_id_studies
cdb_mean_matF
cdb_rbind
cdb_unnest
mat_mean
mpm_mean
mat_to_string
vec_to_string
string_to_mat
string_to_vec
mpm_has_prop
mpm_has_active
mpm_has_dorm
mpm_first_active
Accessor functions
matA
matU
matF
matC
B
Rage
User-defined matrix object
A
=
U
+
F
+
C
Transformation
mpm_split
mpm_rearrange
mpm_collapse
mpm_standardize
standard_stages
repro_stages
Life tables
mpm_to_table
mpm_to_[lx/px/hx/mx]
lx_to_[px/hx]
px_to_[lx/hx]
hx_to_[lx/px]
qsd_converge
Life history traits
life_expect_mean
life_expect_var
longevity
net_repro_rate
gen_time
mature_[age/distrib/prob]
entropy_[k/d]
shape_[rep/surv]
Vital rates
vitalRates
vr_dorm_enter
vr_dorm_exit
vr_fecundity
vr_growth
vr_shrinkage
vr_stasis
vr_survival
vr_vec_...
vr_mat_...
Visualisation
plot_life_cycle
Perturbation analyses
perturb_matrix
perturb_trans
perturb_vr
perturb_stochastic

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 data set for analysis). (**B**) The subsetted 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`), creating of life tables (e.g. `mpm_to_lx`), derivation of life history traits (e.g. `longevity`), calculation of vital rates (e.g. using `vitalRates` to calculate average survival, reproduction, development, etc.), visualisation of life cycles (e.g. `plot_life_cycle`), and perturbation analyses (e.g. `perturb_stochastic`).

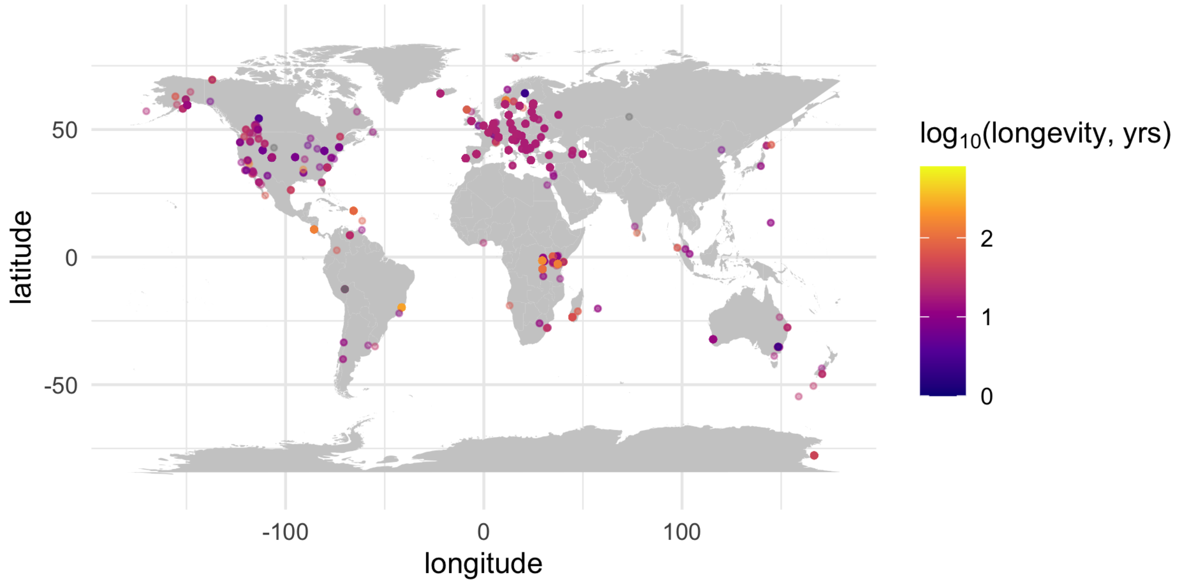

Figure 2. The spatial extent of data in the subset of mammal data used in our example analysis, colour-coded by estimated longevity. Note that 269 of matrices for mammals in our set (~12%) do not have associated spatial information.

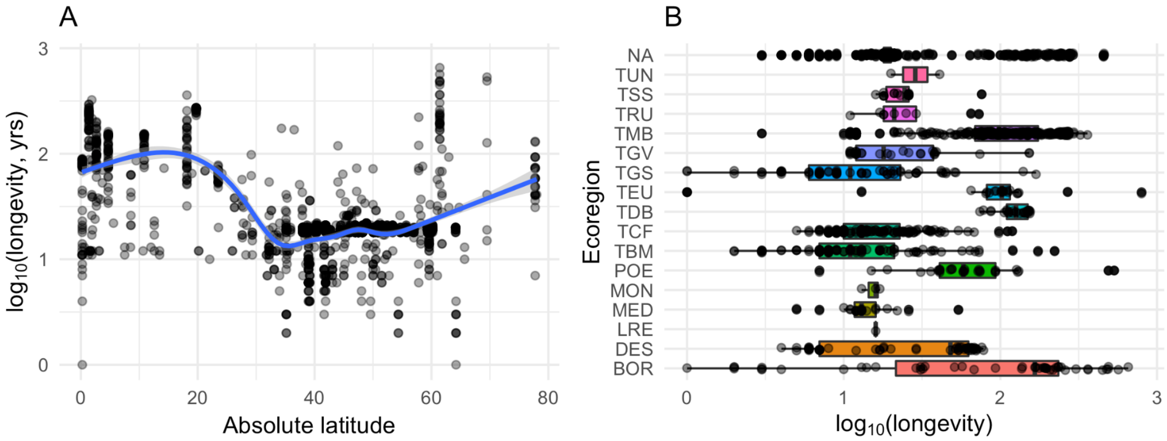

Figure 3. The relationship between estimated longevity and (A) absolute latitude and (B) ecoregion. The line in 3A represents a smoothed spline fitted using `ggplot`'s default settings.

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 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 matrix population models within a `CompadreDB` object, such as missing values, singular **U** matrices, non-ergodicity, non-irreducibility, primitivity etc. (see 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 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 matrix for each set of 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 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 combined (**A**), survival (**U**), sexual reproduction (**F**), and clonal reproduction(**C**) matrices from a `CompadreMat` or `CompadreDB` object.

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 Owen R. 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 (Cochran & Ellner, 1992; Caswell, 2001).
	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 Owen R. 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.
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 standardized 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.