palaeoverse: a community-driven *R* package to support palaeobiological analysis

—

Lewis A. Jones1, William Gearty2, Bethany J. Allen3,4, Kilian Eichenseer5, Christopher D. Dean6, Sofía Galván1, Miranta Kouvari6,7, Emma M. Dunne8, Pedro L. Godoy9,10, Cecily Nicholl6, Lucas Buffan11, Erin M. Dillon12,13, Joseph T. Flannery-Sutherland14, and Alfio Alessandro Chiarenza1

—

*1Centro de Investigación Mariña, Grupo de Ecoloxía Animal, Universidade de Vigo, 36310 Vigo, Spain.*

*2Division of Paleontology, American Museum of Natural History, New York, NY, 10024 USA.*

*3Department of Biosystems Science and Engineering, ETH Zürich, 4058 Basel, Switzerland.*

*4Computational Evolution Group, Swiss Institute of Bioinformatics, 1015 Lausanne, Switzerland.*

*5Department of Earth Sciences, Durham University, South Road, DH1 3LE, Durham, United Kingdom.*

*6Department of Earth Sciences, University College London, Gower Street, WC1E 6BT, London, United Kingdom.*

*7Life Sciences Department, Natural History Museum, Cromwell Road, SW7 5BD, London, United Kingdom.*

*8Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), GeoZentrum Nordbayern, 91054 Erlangen, Germany.*

*9Laboratório de Paleontologia, Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, SP, 14040-901 Brazil.*

*10Department of Anatomical Sciences, Stony Brook University, Stony Brook, NY, 11794 USA.*

*11Département de Biologie, École Normale Supérieure de Lyon, Université Claude Bernard Lyon 1, 69342 Lyon Cedex 07, France.*

*12Smithsonian Tropical Research Institute, Balboa, Republic of Panama.*

*13Department of Ecology, Evolution, and Marine Biology, University of California, Santa Barbara, CA 93106, USA.*

*14School of Earth Sciences, University of Bristol, BS8 1RL, Bristol, UK*

—

**Corresponding author:** LewisAlan.Jones@uvigo.es

Abstract

1. The open-source programming language ‘R’ has become a standard tool in the palaeobiologists’ toolkit. Its popularity amongst the community continues to grow with published articles often citing the usage of R, and R packages. However, while several packages exist to support palaeobiological analysis, there is a lack of agreed standards for data preparation and of packages to support this. Consequently, data preparation steps are often unclear and not reproducible, even when code is provided. Moreover, due to a lack of code accessibility and documentation, palaeobiologists are often forced to ‘reinvent the wheel’ to find solutions to issues already resolved by members of the community.
2. Here, we introduce palaeoverse, a community-driven R package for computational palaeobiological research. The package is freely available and has three core principles: (1) streamline data preparation and analyses; (2) enhance code readability; and (3) improve reproducibility of results. As part of the development of the package, we combined our experience with that of the wider palaeobiological community to understand what tools are needed. The latter was collated by conducting an online survey distributed via social media and email exchange.
3. We first report the findings of the survey which shaped the development of the package. Subsequently, we describe and demonstrate the functionality available in palaeoverse and provide usage examples. Finally, we discuss the resources we have made available for the community, and the future perspectives of the palaeoverse project.
4. palaeoverse is the first community driven R package in palaeobiology, developed with the intention of bringing the community together to establish agreed standards, assisting to modernise the discipline. The package provides a user-friendly platform for preparing data for analysis with well-documented code to enhance transparency. Finally, the functionality available in palaeoverse enables rapid reuse and reproducibility of research results, as well as improving code readability to support the review process.

# Keywords

Analytical Palaeobiology, Computational Palaeobiology, R programming, Reusable, Readable, Reproducible

# Introduction

Since the development of large palaeontological datasets from the 1970s, palaeontologists have increasingly adopted computational approaches to resolve questions about the history of life on Earth. Today, the majority of sub-disciplines within palaeontology regularly require the use of large datasets of various formats to perform experiments *in silico*. This has initiated a ‘Golden Age’ of palaeontology (Sepkoski & Ruse, 2009), where extensive datasets have enabled the resolution of long-standing questions about macroevolution and macroecology (Close et al., 2020; Mannion et al., 2014; Quental & Marshall, 2013; Zaffos et al., 2017). The advancement and increasing availability of datasets has meant that coding is now an integral part of palaeobiology, and is widely used to clean (Flannery-Sutherland, Raja, et al., 2022), analyse (Kocsis et al., 2019), and visualise data (Bell & Lloyd, 2015), as well as build models (Silvestro et al., 2014; Starrfelt & Liow, 2016) and implement simulations (Fraser, 2017; Furness et al., 2021). Whilst software has been developed in languages such as C++ (Garwood et al., 2019) or Python (Silvestro et al., 2014) to support palaeobiological research, the programming language R is currently the most popular language in palaeobiology. This is due to the wide range of tools–in the form of R packages–available to help users analyse their data. Whilst these tools are often borrowed or repurposed from ecology (Chao et al., 2014; Oksanen et al., 2020), there are a variety of packages that have been developed to specifically handle palaeontological data analysis (Kocsis et al., 2019; Lloyd, 2016).

Despite this, there is a lack of packages developed explicitly for preparing data for analyses, forcing users to construct custom scripts. This has three potentially undesirable consequences: (1) unreadable code; (2) non-reusable code; and (3) unreproducible analyses. Palaeobiologists are often not trained programmers, nor should they be expected to be. However, this results in distinct differences in code style and practices amongst the community. While some researchers go to great efforts to improve code legibility and documentation, this is a time consuming task, and understandably not always done. However, in the absence of well-documented and familiar code, custom scripts can be inaccessible (i.e. unreadable) (Filazzola & Lortie, 2022). While increasingly requested by journals, code is not always provided as supplementary material, or made available in online repositories (e.g. GitHub). Consequently, code cannot be reused or adapted by the community, resulting in researchers having to often ‘reinvent the wheel’ (i.e. non-reusable code). Even when code is present, it may have been written in a way that is specific to the dataset being analysed, and might therefore require extensive reworking before it can be applied to other data. In combination, the lack of readable and reusable code leads to research results being unreproducible during the review process, or by future studies aiming to extend the work. Such issues are further exacerbated by an absence of community standards in how data should be prepared for analyses; differing approaches utilised by different researchers result in a lack of consistency between studies, making comparison between results challenging. As these factors greatly hinder the efficiency of computational palaeontological approaches and research, there is an established need for both protocols and tools for preparing palaeontological data for further analysis.

Here, we introduce the R package palaeoverse, a community-driven toolkit for streamlining palaeobiological analyses and improving code readability and reproducibility. Our approach differs from other palaeontological R packages in that it aims to be part of a larger effort to bring the palaeobiology community together. palaeoverse aims to establish consensus on both the steps taken in data preparation for various forms of palaeobiological analysis, and how these steps are implemented. The package contains functions developed in line with community needs to cleanse and prepare occurrence datasets for further analysis. These needs were established by a survey conducted by members of a new working group. The functionality of palaeoverse can be applied to any occurrence dataset, and as such are independent of any external databases. In this paper, we report results from our survey, describe and detail the functionality of palaeoverse, and illustrate its features with usage examples.

# Community survey

We conducted a public survey to collect the opinions of the palaeobiological community, and determine which tools were most needed. The survey was advertised via social media (i.e. Twitter) and a mailing list, and included questions related to researchers’ previous experience, pre-existing code (to identify potential contributions), and what functionality they considered useful for developing for a new package. Survey results indicate the type of data participants typically work with, the tasks commonly carried out when working with this data, and which functionality they would like to have access to in palaeoverse ([Figure 1](#fig-survey)). In general, participants tend to work with a wide range of data ([Figure 1](#fig-survey)), and the checking and transformation of data is the mostly commonly employed task. Various functionality was requested by survey participants, with data plotting, time binning, and data access commonly requested ([Figure 1](#fig-survey)). Finally, over 40% of participants indicated they were willing to contribute code to palaeoverse, reflecting the potential for a community-driven project. Specific details regarding the survey and responses can be found in the Supplementary Material.

|  |
| --- |
| Figure 1: Summary of responses to the palaeoverse survey. (A) Previous experience on the types of palaeontological data that survey participants typically work with. (B) Tasks that respondents routinely carry out in their analyses, and the functions they would find useful in the palaeoverse package. |

# Package description

palaeoverse is a R package that provides a range of auxiliary functions to support data preparation and palaeobiological analysis. A summary of the functions available in palaeoverse are provided in Table 1, and further discussion on functionality is provided within the features section. To demonstrate the functionality and versatility of the package, we also provide usage examples below.

## Installation

The palaeoverse package can be installed from the CRAN using the default install.packages() function in R (R Core Team, 2022):

install.packages("palaeoverse")

If the user prefers, the development version of palaeoverse can also be installed using the R package remotes (Csárdi et al., 2021):

remotes::install\_github("palaeoverse-community/palaeoverse")

Subsequently, palaeoverse can be loaded via the default library() function in R:

library("palaeoverse")

## Data

The palaeoverse package was not developed to provide solutions for any one data source (i.e. the Paleobiology Database or Neotoma). However, functionality is developed around the assumption that most user’s data would be stored in an occurrence dataframe. As such, this is the expected input data format for most functions in palaeoverse, and we avoid unnecessary transformation of data wherever possible. In most instances, the returned object from a function call is also a dataframe, which we believe to be the easiest data structure for most users to understand and work with. This may be undesirable for some advanced users of R. However, we assume these users have the knowledge and capability to transform the data to their desired structure.

## Functions

A summary of the functions available in palaeoverse is provided in [Table 1](#tbl-summary). Detailed descriptions of the functions are provided herein.

Table 1: A summary table of the functions available in the palaeoverse R package.

| **Function** | **Description** |
| --- | --- |
| axis\_geo() | Add an axis of the geological timescale to a plot |
| bin\_lat() | Bin fossil occurrences into latitudinal bins |
| bin\_spatial() | Bin fossil occurrences into spatial bins |
| bin\_time() | Bin occurrences into time bins using different approaches |
| data() | Two example fossil occurrence datasets: ‘tetrapods’ or ‘reefs’ |
| lat\_bins() | Generate latitudinal bins |
| look\_up() | Link user-specified interval names to the International Geological Timescale |
| palaeorotate() | Reconstruct the palaeogeographic distribution of fossil occurrences |
| phylo\_check() | Check taxon names against tips in a phylogeny and/or remove tips from the tree |
| tax\_check() | Check taxon names in a dataset and flag potential issues |
| tax\_range\_geo() | Calculate the geographic range of taxa using different approaches |
| tax\_range\_time() | Calculate and plot the temporal range of taxa |
| tax\_time\_expand() | Convert taxon temporal ranges to interval-level pseudo-occurrences |
| tax\_unique() | Calculate the number of unique taxa in a dataset of occurrences |
| time\_bins() | Generate stratigraphic time bins or near-equal length time bins |

### Example datasets

Two occurrence datasets (tetrapods and reefs) are included in palaeoverse to provide reproducible examples within the function documentation. The tetrapods dataset is a compilation of Carboniferous–Early Triassic tetrapods (*n* = 5,270) from the Paleobiology Database (<https://paleobiodb.org/#/>). The dataset includes variables relevant to common palaeobiological analyses, covering the taxonomic identification of fossils and their geological, geographical and environmental context. The reefs dataset is a compilation of fossil occurrences from Phanerozoic reefs (*n* = 4,363) from the PaleoReefs Database (Kiessling & Krause, 2022). This dataset includes information on the biological, geological, and geographical context of each reef. With the exception of the removal of superfluous columns and the renaming of some columns to improve clarity, both datasets are unaltered from source. Additional information on both datasets can be accessed via ?tetrapods or ?reefs once the package is loaded.

# Load occurrence data  
data("tetrapods")  
data("reefs")

### Time bins

We have developed time\_bins() to enable access to two popular Geological Timescales (GTS): GTS2012 and GTS2020 (Gradstein et al., 2012, 2020). The function allows users to generate temporal bins at different ranks (i.e. stage, epoch, period, era, or eon) for a specified interval input:

# Get stage-level time bins  
time\_bins(interval = "Phanerozoic", rank = "stage", plot = TRUE)

|  |
| --- |
| Figure 2: Phanerozoic stage-level time bins. Plot depicts the uneveness in duration of stratigraphic time bins. Bar colour filling follows the established colour scheme of the International Commision on Stratigraphy (<https://stratigraphy.org/>). |

As evident from the output plot of this function call ([Figure 2](#fig-stages)), temporal bins generated via GTS are inherently uneven in duration. Several previous studies have attempted to circumvent this issue by generating near-equal-length time bins by grouping stages towards a target bin length (Close et al., 2020; Mannion et al., 2015). time\_bins() enables users to generate near-equal-length time bins following this approach ([Figure 3](#fig-equal)) to a specified target size:

# Generate near-equal length time bins  
time\_bins(interval = "Phanerozoic", rank = "stage", size = 15, plot = TRUE)

|  |
| --- |
| Figure 3: Phanerozoic near-equal-length time bins. Plot depicts composite stratigraphic bins (grouping stage-level bins) for the Phanerozoic of a target bin size of 15 million years. **Note:** time bins are still uneven, but less so than stage-level bins. |

However, we do not advocate the use of one approach over another for generating time bins. Near-equal-length time bins can be more even in length–which is perhaps a statistical goal–but this does not ensure that the amount of rock available for sampling in these bins is even. Whilst not a statistician’s dream, GTS bins are defined on observed data and events from the geological record and avoids grouping ‘apples and pears’. Additional functionality in time\_bins() allows the user to assign occurrences to the generated bins if age estimates are known. However, the bespoke bin\_time() function (discussed below) is likely to be the preferred option for most fossil occurrence data.

### Occurrence binning

Fossil occurrences are frequently ‘binned’ into distinct time intervals to enable observation of changes (e.g. biodiversity or disparity) through geological time. The function bin\_time() allows users to assign occurrences into time bins generated by the function time\_bins() or those defined by the user:

# Generate temporal bins  
bins <- time\_bins()  
# Assign occurrences to bins  
bin\_time(occdf = tetrapods, bins = bins, method = "mid")

Whilst binning occurrences with tightly defined temporal limits proves straightforward, those with poorly constrained maximum and minimum ages can span several intervals, and therefore cannot be easily assigned to a single bin. Palaeontologists have identified numerous solutions to address this problem (Davies et al., 2017; Dean et al., 2020; Franeck & Liow, 2020; Lloyd et al., 2012; Silvestro et al., 2016), but there is currently no consensus on the best methodological approach and subsequent implementation. The bin\_time() function provides five potential approaches defined by the ‘method’ argument: ‘mid’ (assigned based on the midpoint of the temporal range of the occurrence), ‘majority’ (assigned to the bin which covers the majority of the temporal range of the occurrence),‘all’ (assigned to all bins within the temporal range of the occurrence), ‘random’, (assigned randomly to bins within the temporal range of the occurrence, repeated up to assigned ‘reps’), and ‘point’ (assigned randomly from a uniform distribution over the temporal range of the occurrence, repeated up to assigned ‘reps’). We hope that formally including these options within the bin\_time() function will encourage palaeontologists to routinely explore and compare the outcomes of various binning approaches with ease.

Palaeobiologists are also often concerned with understanding the latitudinal distribution of biodiversity in deep time (Allen et al., 2020; Jones et al., 2021; Kröger, 2018; Nicholson et al., 2016). To ease implementation of such analyses, we have also developed two functions lat\_bins() and bin\_lat(), which can be used to generate latitudinal bins of a given size, and assign occurrence data to those respective bins.

# Generate latitudinal bins  
bins <- lat\_bins(size = 15)  
# Assign occurrences to bins  
bin\_lat(occdf = tetrapods, bins = bins, lat = "lat")

In recent years, palaeobiology has developed a heightened interest in the spatial structure of the fossil record, with studies focused on understanding the spatial distribution of biodiversity, and the processes that drive them (Antell et al., 2020; Close et al., 2017; Flannery-Sutherland, Silvestro, et al., 2022). In order to support such analyses, bin\_spatial() has been developed for palaeoverse. The function allows the user to assign occurrence data into equal-area grid cells using discrete hexagonal grids via the h3jsr package (O’Brien, 2022). Additional functionality allows simultaneous assignation of occurrence data to cells of a finer-scale (i.e. a ‘sub-grid’) within the primary grid. This might be desirable for users to evaluate the differences in the amount of area occupied by occurrences within their primary grid cells. This functionality also allows the user to easily rarefy across sub-grid cells within primary cells to further standardise spatial sampling.

# Assign data to equal-area spatial bins  
bin\_spatial(occdf = reefs, spacing = 250)  
bin\_spatial(occdf = reefs, spacing = 250, sub\_grid = 50)

### Palaeogeographic reconstruction

Palaeobiologists use plate rotation models to reconstruct the location of fossil occurrences at time of deposition. While some databases (e.g. the Paleobiology Database) provide pre-generated palaeocoordinates for occurrences, they do not contain all published or unpublished data. Moreover, these databases provide reconstructed coordinates from only one or two of the many plate rotation models available, and it is not always transparent which model (or version of the model) has been used. This lack of transparency is reflected in some published articles that only cite the use of the GPlates (Müller et al., 2018) to reconstruct palaeocoordinates, yet lack specifics on which plate rotation model was used with the GPlates web service or desktop software.

We have developed the function palaeorotate to address these shortcomings. The function allows palaeocoordinates to be reconstructed within R using two different approaches (‘point’ and ‘grid’). The first makes use of the GPlates web service, and allows point data to be rotated to specific ages using a range of available models. The second approach uses reconstruction files of pre-generated palaeocoordinates to spatiotemporally link occurrences’ modern coordinates and age estimate with its respective palaeocoordinates. These rotation files were generated using a 1º x 1º spatial grid, and allows palaeocoordinates to be generated efficiently for large datasets. Additional functionality when using the reconstruction files allows the user to calculate the palaeolatitudinal range between reconstructed coordinates, as well as the maximum great circle distance.

# Add midpoint age for rotation  
tetrapods$age <- (tetrapods$max\_ma + tetrapods$min\_ma) / 2  
# Palaeorotate occurrences and return uncertainty  
palaeorotate(occdf = tetrapods, method = "grid", uncertainty = TRUE)

### Taxon-related features

When working with large occurrence datasets, it is easy for errors to creep into the data. One frequently encountered issue is spelling variations of the same taxon name. This can have undesirable consequences when conducting analyses such as calculating taxonomic richness or abundance. We provide the tax\_check function in palaeoverse to enable the user to compute string distances between provided taxonomic names via the heuristic metric Jaro distance (Jaro, 1989). This metric provides a measure of dissimilarity between 0 (exact match) and 1 (completely dissimilar). During function call, the user defines a threshold for string dissimilarity to identify potential synonyms. In tax\_check, Jaro distances are calculated via the stringdistmatrix function from the stringdist package (van der Loo, 2014).

# Check for taxonomic errors  
tax\_check(occdf = tetrapods, name = "genus")

The function tax\_unique is designed to improve the accuracy of richness estimates obtained from fossil occurrence data. Palaeobiologists routinely remove occurrence data not identified to a desired taxonomic resolution. For example, if analysis is conducted at species-level, occurrences identified to genus level (or above) are discarded from the dataset. However, these occurrences can represent unique species, and their removal can impact richness estimation. The tax\_unique function reduces the amount of occurrences being discarded by retaining fossils which are identified to a coarser taxonomic resolution than the desired level, but must represent a clade not already in the filtered dataset. For instance, with three fossil occurrences identified as *Tyrannosaurus rex*, *Spinosaurus aegyptiacus*, and Diplodocidae, the latter would be discarded under species-level analysis (i.e. a species richness of two). However, Diplodocidae clearly represents a different species. Using tax\_unique, Diplodocidae would be treated as an additional species (i.e. a species richness of three). However, the approach is also conservative in that if multiple coarsely identified occurrences exist in the dataset, these are collapsed to the minimum number of possible species (i.e. two occurrences of Diplodocidae would be treated as only one species). This method is similar to the cryptic diversity measure introduced by Mannion et al. (2011).

# Evaluate unique taxa  
tax\_unique(occdf = tetrapods)

Two functions exist in palaeoverse for computing taxon ranges. The first, tax\_range\_time can be called to calculate and plot the temporal range of taxa. That is, the function will identify all unique taxa provided in the occurrence dataframe and find their first and last appearance dates. The second, tax\_range\_geo can be called to calculate the geographic range of taxa. As there is no one single metric for calculating geographic range (Darroch et al., 2020), this function allows the user to specify one of four different approaches: (1) the area of a convex hull; (2) the (palaeo-)latitudinal range; (3) the maximum great-circle distance; and (4) the number of occupied equal-area grid cells. Similar to tax\_range\_time, the function will identify all unique taxa provided, and calculate these metrics based on the available occurrences of each taxa.

# Remove NA data  
tetrapods <- subset(tetrapods, !is.na(order))  
# Compute temporal range of orders  
tax\_range\_time(occdf = tetrapods, name = "order", plot = TRUE)

|  |
| --- |
| Figure 4: Temporal range of tetrapod orders in the palaeoverse example dataset. |

# Compute latitudinal range of orders  
tax\_range\_geo(occdf = tetrapods, name = "order", method = "lat")

The tax\_time\_expand function is provided to convert taxon temporal range data to interval-level pseudo-occurrences. While these pseudo-occurrences should not be treated as equivalent to actual occurrence data, such data can be useful for performing statistical analyses where interval-level data is required.

# Create example dataframe  
taxdf <- data.frame(name = c("A", "B", "C"),  
 max\_ma = c(150, 60, 30),  
 min\_ma = c(110, 20, 0))  
# Convert to interval level pseudo-occurrences  
tax\_time\_expand(taxdf = taxdf)

### Phylogeny wrangling

The function phylo\_check compares a list of taxonomic names to the list of tip names in a user-provided phylogeny. This information can be provided as a table describing the presence or absence of each taxon in the list and/or tips, or as counts of taxa present only in the list, only in the phylogeny, or in both. The function can also be used to trim the phylogeny to only include branches whose tip names are included within the list of taxonomic names.

### Additional features

A common feature request from our survey was the ability to add the ‘Geological Timescale’ to time-series plots in base R, with similar behaviour to the deeptime R package (Gearty, 2022) for ggplot2 (Wickham, 2016). To address this request, the axis\_geo function has been developed for the palaeoverse package and has similar functionality to that of deeptime::coord\_geo ([Figure 5](#fig-axis)).

plot(x = 540:0, xlab = "Time (Ma)", ylab = "User-variable",  
 xlim = c(540, 0), xaxt = "n")  
axis\_geo(side = 1, intervals = "periods")

|  |
| --- |
| Figure 5: Example Phanerozoic plot. Plot depicts the usage of the axis\_geo function for adding the Geological Timescale to a base R plot. |

A common difficulty faced by palaeontologists is that the temporal information related to fossil occurrence data is often asynchronous, and not directly comparable. Temporal data may be provided as either interval names or numeric ages, and might conform to different time scales (e.g. international stages from the International Commision on Stratigraphy, or North American land mammal ages). While interval names tend to be relatively stable over time, numerical age estimates are frequently updated with improved dating techniques, or the collection of new data. Consequently, where possible, interval names should thus be used to correlate occurrences from different stratigraphic time scales. The look\_up function is provided to help assign a common time scale–international stages–to occurrence data. This is achieved with a user-defined table that links arbitrary interval names to corresponding stages on a common time scale. Numerical ages for the assigned stages can be provided by the user, or looked up in GTS2012 or GTS2020.

look\_up(occdf = tetrapods, interval\_key = "example")

# Resources

To support the aims and use of palaeoverse, we have made several resources available to the community. Firstly, we have built a package website (<www.palaeoverse.org>) which provides information on how to contribute to palaeoverse, how to report issues and bugs, and a general community code of conduct. Secondly, a poster and ‘cheat sheet’ is provided via the website which serves as a visual guide on the functionality available in palaeoverse. Finally, a tutorial is provided to demonstrate usage examples with real fossil occurrence data.

# Future perspectives

While the initial development of the palaeoverse R package was led by the authors of this manuscript, it was also informed by the opinions of 34 additional researchers (survey participants). Our hope is that palaeoverse will evolve into a community-driven package with contributions from the wider palaeontological community broadening available functionality. To support this aim, we have provided guidance on how the community can contribute to palaeoverse on the package website (<http://www.palaeoverse.org>). This approach has the wider aims of establishing community standards and consensus in computational palaeobiological research, and enhancing the potential for comparison between studies. Moreover, a community-driven package will assist in making code more familiar and readable to fellow researchers, prevent researchers from reinventing the wheel for common procedures, and improve overall reproducibility of research through the use of computational tools which have been vetted and accepted by the broader community.

The development of the palaeoverse R package marks an initial effort to both streamline palaeontological analysis pipelines and unite the computational palaeobiology community. However, future efforts will see the expansion of the palaeoverse ‘universe’. For example, whilst the functions of palaeoverse are currently focused on handling occurrence based datasets, the scope of the package could be expanded to handle other data types (such as phylogenetic datasets) while maintaining the same base principles of readable, reusable, and reproducible research. We also aim to broaden the reach of palaeoverse beyond a singular R package with the development of additional packages, relevant Shiny applications to support non-R users and teaching exercises, tutorials to offer guidance for new researchers, and workshops to provide practical experience. We hope these efforts foster collaboration and the sharing of resources amongst the community. Finally, we warmly welcome the community to join these efforts and have established a community space accordingly to help facilitate the process (<https://groups.google.com/g/palaeoverse>).

# Acknowledgements

The authors are extremely grateful to all survey respondents who helped to shape the development of palaeoverse. The contributions of LAJ, SG, and AAC were supported by the European Research Council under the European Union’s Horizon 2020 research and innovation program (grant agreement 947921; MAPAS project). The contributions of BJA were supported by an ETH+ grant (BECCY). The contributions of CDD were supported by a Royal Society grant (RF\_ERE\_210013). This is Paleobiology Database publication no XXX. **Please add and update where appropriate :-)**.

# Authors’ contributions

LAJ conceived the project. All authors contributed to developing the project. LAJ, BJA, WG, KE, CD, and JF contributed the code. LAJ, BJA, WG, KE, CD, PG, SG, and LB contributed to testing and reviewing the code. All authors contributed to writing the manuscript. **Please add and update where appropriate, you have all contributed in some way :-)**.

# Data accessibility

The palaeoverse R package is hosted on CRAN (XXX), and is available on GitHub (<https://github.com/palaeoverse-community/palaeoverse>). All example datasets are bundled with the R package.

# References

Allen, B. J., Wignall, P. B., Hill, D. J., Saupe, E. E., & Dunhill, A. M. (2020). The latitudinal diversity gradient of tetrapods across the permo-triassic mass extinction and recovery interval. *Proceedings of the Royal Society B*, *287*(1929), 20201125.

Antell, G. S., Kiessling, W., Aberhan, M., & Saupe, E. E. (2020). Marine biodiversity and geographic distributions are independent on large scales. *Current Biology*, *30*(1), 115–121.e5. https://doi.org/<https://doi.org/10.1016/j.cub.2019.10.065>

Bell, M. A., & Lloyd, G. T. (2015). *Strap: An r package for plotting phylogenies against stratigraphy and assessing their stratigraphic congruence*. Wiley Online Library.

Chao, A., Gotelli, N. J., Hsieh, T. C., Sande, E. L., Ma, K. H., Colwell, R. K., & Ellison, A. M. (2014). Rarefaction and extrapolation with hill numbers: A framework for sampling and estimation in species diversity studies. *Ecological Monographs*, *84*, 45–67.

Close, R. A., Benson, R. B. J., Alroy, J., Carrano, M. T., Cleary, T. J., Dunne, E. M., Mannion, P. D., Uhen, M. D., & Butler, R. J. (2020). The apparent exponential radiation of phanerozoic land vertebrates is an artefact of spatial sampling biases. *Proceedings of the Royal Society B: Biological Sciences*, *287*(1924), 20200372. <https://doi.org/10.1098/rspb.2020.0372>

Close, R. A., Benson, R. B., Upchurch, P., & Butler, R. J. (2017). Controlling for the species-area effect supports constrained long-term mesozoic terrestrial vertebrate diversification. *Nature Communications*, *8*(1), 1–11.

Csárdi, G., Hester, J., Wickham, H., Chang, W., Morgan, M., & Tenenbaum, D. (2021). *Remotes: R package installation from remote repositories, including ’GitHub’*. <https://CRAN.R-project.org/package=remotes>

Darroch, S. A., Casey, M. M., Antell, G. S., Sweeney, A., & Saupe, E. E. (2020). High preservation potential of paleogeographic range size distributions in deep time. *The American Naturalist*, *196*(4), 454–471.

Davies, T. W., Bell, M. A., Goswami, A., & Halliday, T. J. (2017). Completeness of the eutherian mammal fossil record and implications for reconstructing mammal evolution through the cretaceous/paleogene mass extinction. *Paleobiology*, *43*(4), 521–536.

Dean, C. D., Chiarenza, A. A., & Maidment, S. C. (2020). Formation binning: A new method for increased temporal resolution in regional studies, applied to the late cretaceous dinosaur fossil record of north america. *Palaeontology*, *63*(6), 881–901.

Filazzola, A., & Lortie, C. (2022). A call for clean code to effectively communicate science. *Methods in Ecology and Evolution*.

Flannery-Sutherland, J. T., Raja, N. B., Kocsis, Á. T., & Kiessling, W. (2022). Fossilbrush: An r package for automated detection and resolution of anomalies in palaeontological occurrence data. *Methods in Ecology and Evolution*.

Flannery-Sutherland, J. T., Silvestro, D., & Benton, M. J. (2022). Global diversity dynamics in the fossil record are regionally heterogeneous. *Nature Communications*, *13*(1), 1–17.

Franeck, F., & Liow, L. H. (2020). Did hard substrate taxa diversify prior to the great ordovician biodiversification event? *Palaeontology*, *63*(4), 675–687.

Fraser, D. (2017). Can latitudinal richness gradients be measured in the terrestrial fossil record? *Paleobiology*, *43*(3), 479–494.

Furness, E. N., Garwood, R. J., Mannion, P. D., & Sutton, M. D. (2021). Evolutionary simulations clarify and reconcile biodiversity-disturbance models. *Proceedings of the Royal Society B*, *288*(1949), 20210240.

Garwood, R. J., Spencer, A. R., & Sutton, M. D. (2019). RE voSim: Organism-level simulation of macro and microevolution. *Palaeontology*, *62*(3), 339–355.

Gearty, W. (2022). *Deeptime: Plotting tools for anyone working in deep time*. <https://CRAN.R-project.org/package=deeptime>

Gradstein, F. M., Ogg, J. G., Schmitz, M. D., & Ogg, G. M. (2020). *Geologic time scale 2020*. Elsevier.

Gradstein, F. M., Ogg, J. G., Schmitz, M., & Ogg, G. (2012). *The geologic time scale 2012*. elsevier.

Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, *84*(406), 414–420.

Jones, L. A., Dean, C. D., Mannion, P. D., Farnsworth, A., & Allison, P. A. (2021). Spatial sampling heterogeneity limits the detectability of deep time latitudinal biodiversity gradients. *Proceedings of the Royal Society B*, *288*(1945), 20202762.

Kiessling, W., & Krause, C. (2022). *PaleoReefs database (PARED)* (Version 1.0) [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.6037852>

Kocsis, Á. T., Reddin, C. J., Alroy, J., & Kiessling, W. (2019). The r package divDyn for quantifying diversity dynamics using fossil sampling data. *Methods in Ecology and Evolution*, *10*(5), 735–743.

Kröger, B. (2018). Changes in the latitudinal diversity gradient during the great ordovician biodiversification event. *Geology*, *46*(2), 127–130.

Lloyd, G. T. (2016). Estimating morphological diversity and tempo with discrete character-taxon matrices: Implementation, challenges, progress, and future directions. Biological journal of the linnean society. *Biological Journal of the Linnean Society*, *118*, 131–151.

Lloyd, G. T., Pearson, P. N., Young, J. R., & Smith, A. B. (2012). Sampling bias and the fossil record of planktonic foraminifera on land and in the deep sea. *Paleobiology*, *38*(4), 569–584.

Mannion, P. D., Benson, R. B., Carrano, M. T., Tennant, J. P., Judd, J., & Butler, R. J. (2015). Climate constrains the evolutionary history and biodiversity of crocodylians. *Nature Communications*, *6*(1), 1–9.

Mannion, P. D., Upchurch, P., Benson, R. B., & Goswami, A. (2014). The latitudinal biodiversity gradient through deep time. *Trends in Ecology & Evolution*, *29*(1), 42–50.

Mannion, P. D., Upchurch, P., Carrano, M. T., & Barrett, P. M. (2011). Testing the effect of the rock record on diversity: A multidisciplinary approach to elucidating the generic richness of sauropodomorph dinosaurs through time. *Biological Reviews*, *86*(1), 157–181. https://doi.org/<https://doi.org/10.1111/j.1469-185X.2010.00139.x>

Müller, R. D., Cannon, J., Qin, X., Watson, R. J., Gurnis, M., Williams, S., Pfaffelmoser, T., Seton, M., Russell, S. H. J., & Zahirovic, S. (2018). GPlates: Building a virtual earth through deep time. *Geochemistry, Geophysics, Geosystems*, *19*(7), 2243–2261. https://doi.org/<https://doi.org/10.1029/2018GC007584>

Nicholson, D. B., Holroyd, P. A., Valdes, P., & Barrett, P. M. (2016). Latitudinal diversity gradients in mesozoic non-marine turtles. *Royal Society Open Science*, *3*(11), 160581.

O’Brien, L. (2022). *h3jsr: Access uber’s H3 library*. <https://CRAN.R-project.org/package=h3jsr>

Oksanen, J., Blanchet, F. G., Friendly, M., Kindt, R., Legendre, P., McGlinn, D., Minchin, P. R., O’Hara, R. B., Simpson, G. L., Solymos, P., Stevens, M. H. H., Szoecs, E., & Wagner, H. (2020). *Vegan: Community ecology package*. <https://CRAN.R-project.org/package=vegan>

Quental, T. B., & Marshall, C. R. (2013). How the red queen drives terrestrial mammals to extinction. *Science*, *341*(6143), 290–292.

R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>

Sepkoski, D., & Ruse, M. (2009). *The paleobiological revolution: Essays on the growth of modern paleontology*. University of Chicago Press.

Silvestro, D., Salamin, N., & Schnitzler, J. (2014). PyRate: A new program to estimate speciation and extinction rates from incomplete fossil data. *Methods in Ecology and Evolution*, *5*(10), 1126–1131.

Silvestro, D., Zizka, A., Bacon, C. D., Cascales-Minana, B., Salamin, N., & Antonelli, A. (2016). Fossil biogeography: A new model to infer dispersal, extinction and sampling from palaeontological data. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *371*(1691), 20150225.

Starrfelt, J., & Liow, L. H. (2016). How many dinosaur species were there? Fossil bias and true richness estimated using a poisson sampling model. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *371*(1691), 20150219.

van der Loo, M. P. J. (2014). The stringdist package for approximate string matching. *The R Journal*, *6*, 111–122. <https://CRAN.R-project.org/package=stringdist>

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>

Zaffos, A., Finnegan, S., & Peters, S. E. (2017). Plate tectonic regulation of global marine animal diversity. *Proceedings of the National Academy of Sciences*, *114*(22), 5653–5658.