A Julia toolkit for species distribution data

Timothée Poisot [1], Ariane Bussières-Fournel [1], Gabriel Dansereau [1] and Michael D. Catchen [1]

[1] Université de Montréal

***Abstract*:** (1) Species distribution modeling requires to handle varied types of data, and benefits from an integrated approach to programming. (2) We introduce **SpeciesDistributionToolkit**, a **Julia** package aiming to facilitate the production of species distribution models. It covers various steps of the data collection and analysis process, extending to the development of interfaces for integration of additional functionalities. (3) By relying on semantic versioning and strong design choices on modularity, we expect that this package will lead to improved reproducibility and long-term maintainability. (4) We illustrate the functionalities of the package through several case studies, accompanied by reproducible code.

***Keywords*:** species distribution models, biogeography, occurrence data, land use, climatic data, pseudo-absences

Introduction

Species Distribution Models [SDMs; Elith and Leathwick (2009)], in addition to being key tools to further our knowledge of biodiversity, are key components of effective conservation decisions (Guisan et al. 2013), planning (McShea 2014), and ecological impact assesment (Baker et al. 2021). The training and evaluation of a SDM is a complex process, with key decisions to make on design and reporting (Zurell et al. 2020). The ability to link data to these steps is central to support the correct interpretation of these models (Araújo et al. 2019). In the recent years, there has been an increase in the number of software packages and tools to assist ecologists with various steps of the development of species distribution models.

As Kass et al. (2024) point out, this increase in the diversity of software tools (most of them in the **R** language) is a good thing. Because the SDMs are a general-purpose methodology, a varied software offers increases the chances that specific decisions can be chained together in the way that best support a specific use case. By making code available for all users, package developers reduce the need for custom implementation of analytical steps, and contribute to the adoption of good practices in the field. However, because building, validating, and applying SDMs requires a diversity of data types, from different sources, many existing packages have been designed independentl. Therefore, they may suffer from low interoperability, which can create friction when using multiple tools together. As an illustration, Kellner et al. (2025) highlight that about 20% of publications for abundance or distribution models are not reproducible because of issues in package dependencies.

To promote interoperability and improve reproductibility, tools that provide an integrated environment are important. In this manuscript, we present **SpeciesDistributionToolkit** (abbreviated as **SDT**), a meta-package for the **Julia** programming language, offering an integrated environment for the retrieval, formatting, and interpretation of data relevant to the modeling of species distributions. **SDT** was in part designed to work within the BON-in-a-Box project (Gonzalez et al. 2023, Griffith et al. 2024), a GEO BON initiative to facilitate the calculation and reporting of biodiversity indicators supporting the Kunming-Montréal Global Biodiversity Framework. A leading design consideration for **SDT** was therefore to maximize interoperability between components and functionalities from the ground up. This is achieved through two mechanisms. First, by relying on strict semantic versioning: package releases provide information about the compatibility of existing code. Second, through the use of *interfaces*: separate software components (including ones external to the package) can interact without prior knowledge of either implementation, and without *dependencies* between the components of **SDT**.

In this manuscript, we describe provide a high-level overview of the functionalities of the package(s) forming **SDT**. We then discuss design principles that facilitate long-term maintenance, development, and integration. We finish by presenting four illustrative case studies: extraction of data at known species occurrences, manipulation of multiple geospatial layers, training and explanation of a SDM, and creation of a virtual species. This later case study is intended to provide an impression of what using **SDT** as a support for the development of novel analyses feels like. All of the case studies are available as supplementary material, in the form of fully reproducible, self-contained Jupyter notebooks.

Application description

SpeciesDistributionToolkit is released as a package for the **Julia** programming language (Bezanson et al. 2017). It is licensed under the open-source initiative approved MIT license. It has evolved from a previous collection of packages to handle GBIF and raster data (Dansereau and Poisot 2021), and now provides extended functionalities as well as improved performance. The package is registered in the **Julia** package repository and can be downloaded and installed anonymously. It is compatible with the current long-term support (LTS) release of **Julia**. The full source code, complete commit history, plans for future development, and a forum, are available at `https://github.com/PoisotLab/SpeciesDistributionToolkit.jl`. This page additionally has a link to the documentation, containing a full reference for the package functions, a series of briefs how-to examples, and longer vignettes showcasing more integrative tutorials.

Component packages

An overview of the **SDT** package is given in Figure 1. The project is organized as a "monorepo", in which multiple separate, but interoperable, packages live. This allows expanding the scope of the package by moving functionalities into new component packages, without complicating the installation process. As **SDT** is registered in the **Julia** package repository, it can be installed by using `add SpeciesDistributionToolkit` when in package mode at the **Julia** prompt.

When loading the **SDT** package with `using SpeciesDistributionToolkit`, all component packages are automatically and transparently loaded. Therefore, users do not need to know where a specific method or function resides to use it. In the next section, we discuss how this modular design ensures that we can grow the functionality of the toolkit over time, while maintaining strict backward compatibility with earlier versions *and* allowing full reproducibility of an analysis.

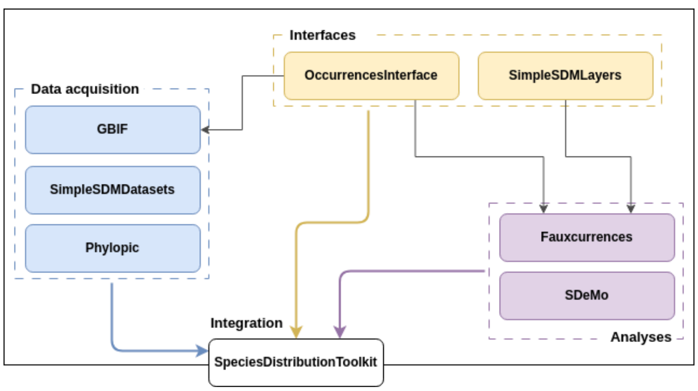

Figure 1: Overview of the packages included in **SpeciesDistributionToolkit**. The packages are color-coded by intended use, and their more specific content is presented in the main text. Note that because the package relies on *interfaces* to facilitate code interoperability, there are only three dependency relationships (black arrows).

The **SDT** package primarily provides integration between the other packages via method overloading (reusing method names for intuitive and concise code), allowing to efficiently join packages together (Roesch et al. 2023). Additional functionalities that reside in the top-level package are the generation of pseudo-absences inspired by Barbet-Massin et al. (2012), access to the `gadm.org` database, handling of polygon data and zonal statistics, and various quality of life methods. Because of the modular nature of the code, any of these functions can be transparently moved to their own packages in the future without affecting reproducibility.

The **SimpleSDMLayers** package offers a series of types to represent raster data in various projections. This package provides the main data representation for most spatial functionalities that **SDT** supports, and handles saving and loading data. In addition, this package contains a number of utility functions to deal with raster data, including interpolation to different spatial grids and CRS, rescaling and quantization of data, and most mathematical operations that can be applied to rasters.

The **OccurrencesInterface** is a light-weight package to provide a common interface for occurrence data. It implements abstract and concrete types to define a single occurrence and a collection thereof, and a series of methods allowing any occurrence data provider (e.g. GBIF) or data representation to become fully interoperable with the rest of **SDT**. All **SDT** methods that handle occurrence data do so through the interface provided by the **OccurrencesInterface** package, allowing future data sources to be integrated without the need for new code.

The **GBIF** package offers access to the `gbif.org` streaming API (GBIF: The Global Biodiversity Information Facility 2025), including the ability to retrieve, filter, and restart downloads. Although this package provides a rich data representation for occurrence data when access to the full GBIF data schema is required, all the objects it returns adhere to the **OccurrencesInterface** interface.

SimpleSDMDatasets implements an interface to retrieve and locally store raster data, which can be extended by users to support additional data sources. In addition, it offers access to a series of common data sources for spatial biodiversity modeling, including the biodiversity mapping project (Jenkins et al. 2013), the EarthEnv collection for land cover (Tuanmu and Jetz 2014) and habitat heterogeneity (Tuanmu and Jetz 2015), Copernicus land cover 100m data (Buchhorn et al. 2020), the PaleoClim (Brown et al. 2018) data, the WorldClim 1 and 2 data (Fick and Hijmans 2017) and their projections under various RCPs and SSP, and part of the CHELSA 1 and 2 data (Karger et al. 2017) and their projections under various RCPs and SSPs.

Phylopic offers a wrapper around the `phylopic.org` API to download silhouettes for taxonomic entities. It also provides utilities for citation of the downloaded images. Its functionalities are similar to the **rphylopic** package (Gearty and Jones 2023).

The **Fauxcurrences** packages is inspired by the work of Osborne et al. (2022), and allows generating a series of simulated occurrence data that have the same statistical structure as observed ones. The package supports multi-species data, with user-specified weights for conserving intra and inter-specific occurrence distances.

The **SDeMo** package is aimed at providing tools to use as part of training and education material on species distribution modeling. By providing a series of data transformation (PCA, Whitening, z-score) and classifiers (BIOCLIM, Naive Bayes, logistic regression, and decision trees), it offers the basic elements to demonstrate training and evaluation of SDMs, as well as techniques related to heterogeneous ensembles and bagging with support for arbitrary consensus (Marmion et al. 2009) and voting (Drake 2014) functions. In addition, **SDeMo** promotes the use of interpretable techniques: the package supports regular (Elith et al. 2005) and inflated (Zurell et al. 2012) partial responses, as well as the calculation and mapping of Shapley values (Mesgaran et al. 2014, Wadoux et al. 2023) using the standard Monte-Carlo approach (Mitchell et al. 2021). Counterfactuals (Karimi et al. 2019, Van Looveren and Klaise 2019), representing perturbation of the input data leading to the opposite prediction (*i.e.* "what environmental conditions would lead to the species being absent") can also be generated.

Software information

SDT uses the built-in **Julia** package manager to ensure that the version of all dependencies are kept up to date. Furthermore, we use strict semantic versioning: major versions correspond to no breaking changes in user-developped code, minor versions increase with additional functionalities, and patch releases cover minor bug fixes or documentation changes. All packages have a *CHANGELOG* file, which documents what changes are included in each release. Following a constructive cost model analysis (Kemerer 1987) of the version described in this publication, the package represents approx. 11k lines of active code (no blank lines, no comments), for an estimated development cost of approx. 325k USD.

This strict reliance on semantic versioning solves the issues of maintaining compatibility when new functionalities are added: all releases in the *v1.x.x* branch of **SDT** depend on component packages in their respective *v1.x.x* branch, and users can benefit from now functionalities without risking to break existing code. This behavior is extensively tested, both using unit tests, and through integration testing generated as part of the online documentation.

Integration with other packages

The **SDT** package benefits from close integration with other packages in the **Julia** universe. Notably, this includes **Makie** (and all related backends, with support for **GeoMakie**) (Danisch and Krumbiegel 2021) for plotting and interactive data visualisation, where usual plot types are overloaded for both layer and occurrence data. Most data handled by **SDT** can be exported using the **Tables** interface, which allows data to be consumed by other packages like **DataFrames** (Bouchet-Valat and Kamiński 2023) and **MLJ** (Blaom et al. 2020), or directly saved as csv files.

Interfaces to internal **Julia** methods are also implemented whenever they are pertinent. In particular, **SimpleSDMLayers** objects behave like arrays, are iterable, and broadcastable; objects from **OccurrencesInterface** behave as arrays and are similarly iterable. The **SDeMo** package relies on part of the **StatsAPI** interface, allowing to easily define new data transformation and classifier types to support additional features.

Achieving integration with other packages through method overloading and the adherence to well-established interfaces is important, as it increases the chances that additional functionalities external to **SDT** can be used directly or fully supported with minimal addition of code.

Illustrative case studies

In this section, we provide a series of case studies, meant to illustrate the use of the package. The on-line documentation offers longer tutorials, as well as a series of how-to vignettes to illustrate the full scope of what the package allows. The code for each of these case studies is available as fully independent Jupyter notebooks, forming the supplementary material of this article. The example we use throughout is the distribution of *Akodon montensis* (Rodentia, family Cricetidae), a known host of orthohantaviruses (Owen et al. 2010, Burgos et al. 2021), in Paraguay. As the notebooks accompanying this article cover the full code required to run these case studies, we do not present code snippets in the main text, and instead focus on explaining which component packages are used in each example.

Using data from GBIF

To illustrate the interactions between the component packages, we provide a simple illustration (Supp. Mat. 1) where we (i) request occurrence data using the **GBIF** package, (ii) download the silhouette of the species through **Phylopic**, and (iii) extract temperature and precipitation data at the points of occurrence. The results are presented in Figure 2. The full notebook includes information about basic operations on raster data, as well as extraction of data based on occurrence records.

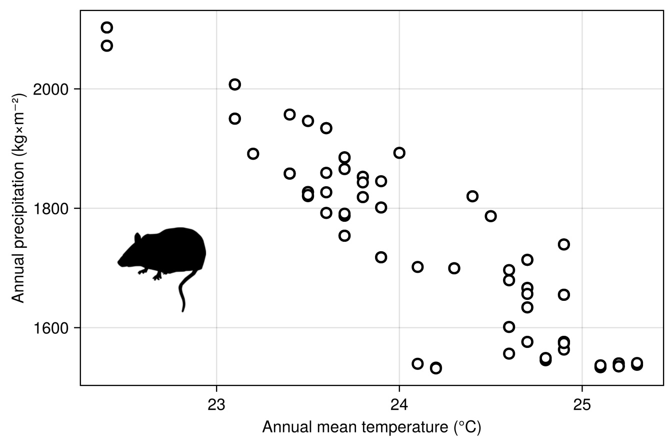

Figure 2: Relationship between temperature and precipitation (BIO1 and BIO12) at each georeferenced occurrence known to GBIF for *Akodon montensis*. The code to produce this figure is available as Supp. Mat. 1.

In practice, although the data are retrieved using the **GBIF** package, they are used internally by **SDT** through the **OccurrencesInterface** package. This package defines a small convention to handle georeferenced occurrence data, and allows to transparently integrate additional occurrence sources. By defining five methods for a custom data type, users can plug-in any occurrence data source and enjoy full compatibility with the entire **SDT** functionalities.

Landcover consensus map

In this case study (Supp. Mat. 2), we retrieve the land cover data from Tuanmu and Jetz (2014), clip them to a GeoJSON polygon describing the country of Paraguay (**SDT** can download data directly from `gadm.org`), and apply the `mosaic` operation to figure out which class is the most locally abundant. This case study uses the **SimpleSDMDatasets** package to download (and locally cache) the raster data, as well as the **SimpleSDMLayers** package to provide basic utility functions on raster data. The results are presented in Figure 3.

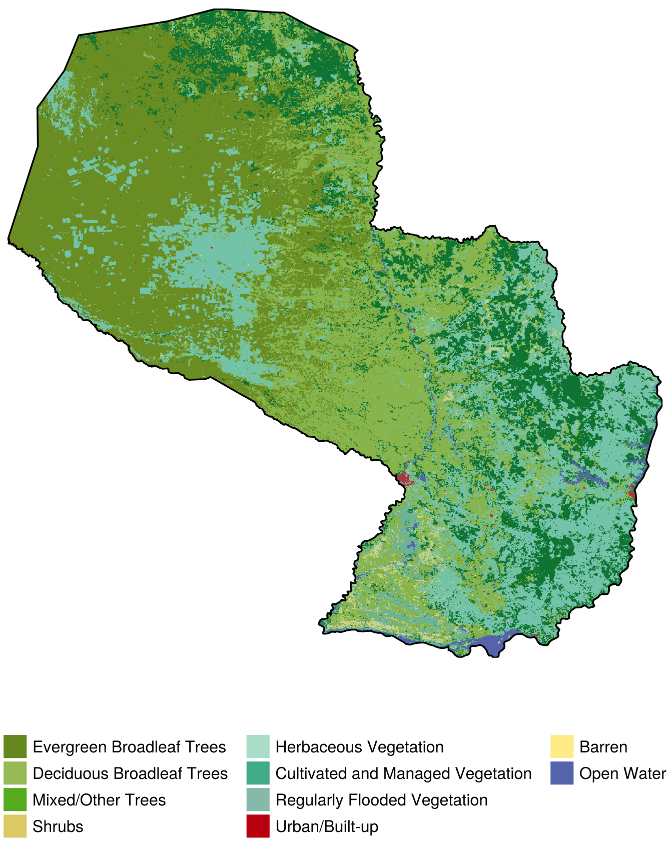

Figure 3: Land cover consensus (defined as the class with the strongest local representation) in the country of Paraguay. Only the classes that were most abundant in at least one pixel are represented. The code to produce this figure is available as Supp. Mat. 2.

When first downloading data through **SimpleSDMDatasets**, they will be stored locally for future use. When the data are requested a second time, they are read directly from the disk, speeding up the process massively. Note that the location of the data is (i) standardized by the package itself, making the file findable to humans, and (ii) changeable by the user to, *e.g.*, store the data within the project folder rather than in a central location. As much as possible, **SDT** will only read the part of the raster data that is required given the region of interest to the user. This is done by providing additional context in the form of a bounding box (in WGS84, regardless of the underlying raster data projection). **SDT** has methods to calculate the bounding box for all the objects it supports.

Training a species distribution model

In this case study, we illustrate the integration of **SDeMo** and **SimpleSDMLayers** to train a species distribution model. We specifically train a rotation forest (Bagnall et al. 2018), an homogeneous ensemble of PCA followed by decision trees. The results are presented in Figure 4. The model is built by selecting an optimal suite of BioClim variables, then predicted in space, and the resulting predicted species range is finally clipped by the elevational range observed in the occurrence data.

Figure 4: Predicted range of *Akodon montensis* in Paraguay based on a rotation forest trained on GBIF occurrences and the BioClim variables. The predicted range is clipped to the elevational range of the species. The code to produce this figure is available as Supp. Mat. 3.

The full notebook (Supp. Mat. 3) has additional information on routines for variable selection, stratified cross-validation, as well as the construction of the ensemble from a single PCA and decision tree. In addition, we report in Figure 5 the partial and inflated partial responses to the most important variable (highlighting an interpretable effect of the variable in the model), as well as the (Monte-Carlo) Shapley values (Mesgaran, Cousens, and Webber 2014, Mitchell, Cooper, Frank, and Holmes 2021, Wadoux, Saby, and Martin 2023) for each prediction in the training set. Because **SDeMo** works through generic functions, these methods can be applied to any model specified by the user. In practice, flexible ML frameworks exist for **Julia**, notably **MLJ** (Blaom, Kiraly, Lienart, Simillides, Arenas, and Vollmer 2020), which can be used for real-world applications.

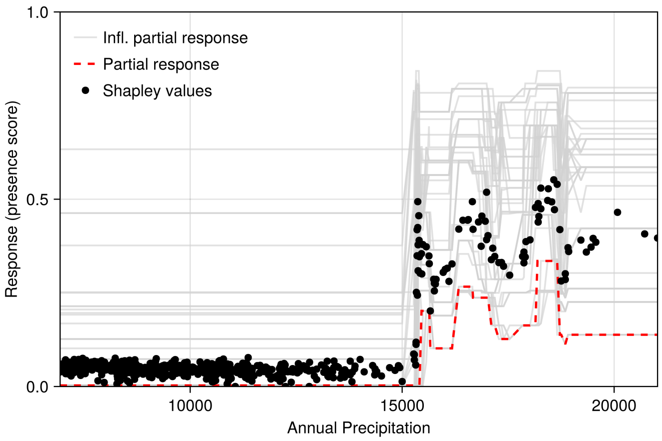

Figure 5: Partial responses (red) and inflated partial responses (grey) to the most important variable. In addition, the Shapley values for all training data are presented in the same figure. Shapley values were added to the average model prediction to be comparable to partial responses. The code to produce this figure is available as Supp. Mat. 3.

Distribution of a virtual species

In the final case study (Supp. Mat. 4), we simulate a virtual distribution (Hirzel et al. 2001), using a species with a logistic response to each environmental covariate (Leroy et al. 2016), and a prevalence similar to the one predicted in Figure 4. The results are presented in Figure 6.

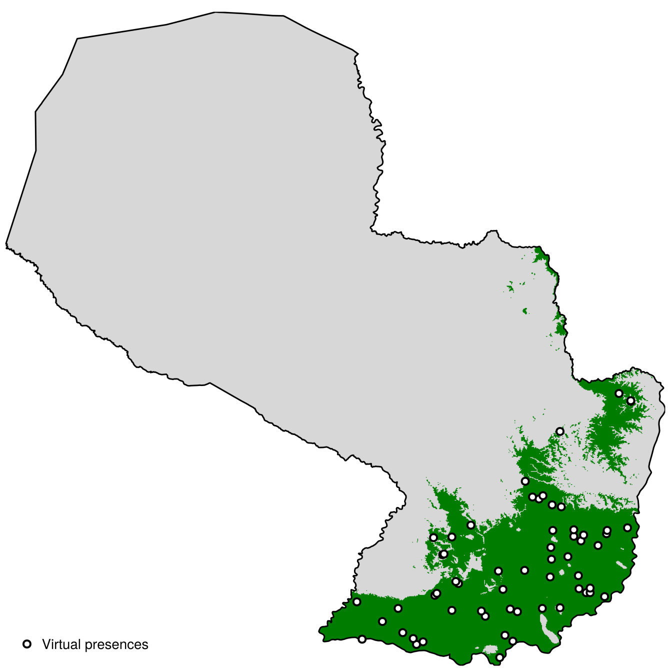

Figure 6: Virtual distribution for a hypothetical species with logistic response to the environment, as well as a sample of simulated occurrences. The prevalence of the virtual species is equivalent to the results in Figure 4. The code to produce this figure is available as Supp. Mat. 4.

Because the layers used by **SDT** are broadcastable, we can rapidly apply a function (here, the logistic response to the environmental covariate) to each layer, and then multiply the suitabilities together. The last step is facilitated by the fact that most basic arithmetic operations are defined for layers, allowing for example to add, multiply, substract, and divide them by one another.

Conclusion

We have presented **SpeciesDistributionToolkit**, a package for the **Julia** programming language aiming to facilitate the collection, curation, analysis, and visualisation of data commonly used in species distribution modeling. Through the use of interfaces and a modular design, we have made this package robust to changes, easy to add functionalities to, and well integrated to the rest of the **Julia** ecosystem. All code for the case studies can be found in Supp. Mat. 1-4.

Plans for active development of the package are focused on (i) additional techniques for pseudo-absence generations, likely leading to their separate component package, (ii) full compatibility with the **MultivariateStatistics** and **Clustering** packages for transformation and aggregation, and (iii) additional **SDeMo** functionalities to allow cross-validation techniques with biologically relevant structure (Roberts et al. 2017).

Acknowledgements: TP is funded by an NSERC Discovery grant, a Discovery Acceleration Supplement grant, and a Wellcome Trust grant (223764/Z/21/Z). MDC is funded by an IVADO Postdoctoral Fellowship.

Bibliography

Araújo M B, Anderson R P, Márcia Barbosa A, et al (2019) Standards for distribution models in biodiversity assessments. Science advances 5:eaat4858. https://doi.org/10.1126/sciadv.aat4858

Bagnall A, Flynn M, Large J, et al (2018) Is rotation forest the best classifier for problems with continuous features?. arXiv [csLG]

Baker D J, Maclean I M D, Goodall M, Gaston K J (2021) Species distribution modelling is needed to support ecological impact assessments. The journal of applied ecology 58:21–26. https://doi.org/10.1111/1365-2664.13782

Barbet-Massin M, Jiguet F, Albert C H, Thuiller W (2012) Selecting pseudo-absences for species distribution models: how, where and how many?: How to use pseudo-absences in niche modelling?. Methods in ecology and evolution 3:327–338. https://doi.org/10.1111/j.2041-210x.2011.00172.x

Bezanson J, Edelman A, Karpinski S, Shah V B (2017) Julia: A fresh approach to numerical computing. SIAM review Society for Industrial and Applied Mathematics 59:65–98. https://doi.org/10.1137/141000671

Blaom A, Kiraly F, Lienart T, et al (2020) MLJ: A Julia package for composable machine learning. Journal of open source software 5:2704–2705. https://doi.org/10.21105/joss.02704

Bouchet-Valat M, Kamiński B (2023) DataFrames.Jl: Flexible and fast tabular data in \textit{Julia}. Journal of statistical software 107:. https://doi.org/10.18637/jss.v107.i04

Brown J L, Hill D J, Dolan A M, et al (2018) PaleoClim, high spatial resolution paleoclimate surfaces for global land areas. Scientific data 5:180254–180255. https://doi.org/10.1038/sdata.2018.254

Buchhorn M, Smets B, Bertels L, et al (2020) Copernicus Global Land Service: Land Cover 100m: collection 3: epoch 2019: Globe

Burgos E F, Vadell M V, Bellomo C M, et al (2021) First evidence of Akodon-borne orthohantavirus in northeastern Argentina. EcoHealth 18:429–439. https://doi.org/10.1007/s10393-021-01564-6

Danisch S, Krumbiegel J (2021) Makie.jl: Flexible high-performance data visualization for Julia. Journal of open source software 6:3349–3350. https://doi.org/10.21105/joss.03349

Dansereau G, Poisot T (2021) SimpleSDMLayers.Jl and GBIF.Jl: A framework for species distribution modeling in Julia. Journal of open source software 6:2872–2873. https://doi.org/10.21105/joss.02872

Drake J M (2014) Ensemble algorithms for ecological niche modeling from presence-background and presence-only data. Ecosphere (Washington, DC) 5:1–16. https://doi.org/10.1890/es13-00202.1

Elith J, Ferrier S, Huettmann F, Leathwick J (2005) The evaluation strip: A new and robust method for plotting predicted responses from species distribution models. Ecological modelling 186:280–289. https://doi.org/10.1016/j.ecolmodel.2004.12.007

Elith J, Leathwick J R (2009) Species distribution models: Ecological explanation and prediction across space and time. Annual review of ecology, evolution, and systematics 40:677–697. https://doi.org/10.1146/annurev.ecolsys.110308.120159

Fick S E, Hijmans R J (2017) WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas: NEW CLIMATE SURFACES FOR GLOBAL LAND AREAS. International journal of climatology: a journal of the Royal Meteorological Society 37:4302–4315. https://doi.org/10.1002/joc.5086

GBIF: The Global Biodiversity Information Facility (2025) \textit{What is GBIF?}

Gearty W, Jones L A (2023) rphylopic: An R package for fetching, transforming, and visualising PhyloPic silhouettes. Methods in ecology and evolution 14:2700–2708. https://doi.org/10.1111/2041-210x.14221

Gonzalez A, Vihervaara P, Balvanera P, et al (2023) A global biodiversity observing system to unite monitoring and guide action. Nature ecology & evolution 1–5. https://doi.org/10.1038/s41559-023-02171-0

Griffith J, Lord J-M, Catchen M D, et al (2024) BON in a Box: An Open and Collaborative Platform for Biodiversity Monitoring, Indicator Calculation, and Reporting. https://doi.org/10.32942/X2M320

Guisan A, Tingley R, Baumgartner J B, et al (2013) Predicting species distributions for conservation decisions. Ecology letters 16:1424–1435. https://doi.org/10.1111/ele.12189

Hirzel A H, Helfer V, Metral F (2001) Assessing habitat-suitability models with a virtual species. Ecological modelling 145:111–121. https://doi.org/10.1016/s0304-3800(01)00396-9

Jenkins C N, Pimm S L, Joppa L N (2013) Global patterns of terrestrial vertebrate diversity and conservation. Proceedings of the National Academy of Sciences of the United States of America 110:E2602–10. https://doi.org/10.1073/pnas.1302251110

Karger D N, Conrad O, Böhner J, et al (2017) Climatologies at high resolution for the earth's land surface areas. Scientific data 4:170122–170123. https://doi.org/10.1038/sdata.2017.122

Karimi A-H, Barthe G, Balle B, Valera I (2019) Model-agnostic counterfactual explanations for consequential decisions. arXiv [csLG]

Kass J M, Smith A B, Warren D L, et al (2024) Achieving higher standards in species distribution modeling by leveraging the diversity of available software. Ecography. https://doi.org/10.1111/ecog.07346

Kellner K F, Doser J W, Belant J L (2025) Functional R code is rare in species distribution and abundance papers. Ecology 106:e4475. https://doi.org/10.1002/ecy.4475

Kemerer C F (1987) An empirical validation of software cost estimation models. Communications of the ACM 30:416–429. https://doi.org/10.1145/22899.22906

Leroy B, Meynard C N, Bellard C, Courchamp F (2016) virtualspecies, an R package to generate virtual species distributions. Ecography 39:599–607. https://doi.org/10.1111/ecog.01388

Marmion M, Parviainen M, Luoto M, et al (2009) Evaluation of consensus methods in predictive species distribution modelling. Diversity & distributions 15:59–69. https://doi.org/10.1111/j.1472-4642.2008.00491.x

McShea W J (2014) What are the roles of species distribution models in conservation planning?. Environmental conservation 41:93–96. https://doi.org/10.1017/s0376892913000581

Mesgaran M B, Cousens R D, Webber B L (2014) Here be dragons: a tool for quantifying novelty due to covariate range and correlation change when projecting species distribution models. Diversity & distributions 20:1147–1159. https://doi.org/10.1111/ddi.12209

Mitchell R, Cooper J, Frank E, Holmes G (2021) Sampling Permutations for Shapley Value Estimation. arXiv [statML]

Osborne O G, Fell H G, Atkins H, et al (2022) Fauxcurrence: simulating multi-species occurrences for null models in species distribution modelling and biogeography. Ecography 2022:e5880. https://doi.org/10.1111/ecog.05880

Owen R D, Goodin D G, Koch D E, et al (2010) Spatiotemporal variation in Akodon montensis (Cricetidae: Sigmodontinae) and hantaviral seroprevalence in a subtropical forest ecosystem. Journal of Mammalogy 91:467–481. https://doi.org/10.1644/09-MAMM-A-152.1

Roberts D R, Bahn V, Ciuti S, et al (2017) Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography 40:913–929. https://doi.org/10.1111/ecog.02881

Roesch E, Greener J G, MacLean A L, et al (2023) Julia for biologists. Nature methods 20:655–664. https://doi.org/10.1038/s41592-023-01832-z

Tuanmu M-N, Jetz W (2014) A global 1-km consensus land-cover product for biodiversity and ecosystem modelling: Consensus land cover. Global ecology and biogeography: a journal of macroecology 23:1031–1045. https://doi.org/10.1111/geb.12182

Tuanmu M-N, Jetz W (2015) A global, remote sensing-based characterization of terrestrial habitat heterogeneity for biodiversity and ecosystem modelling: Global habitat heterogeneity. Global ecology and biogeography: a journal of macroecology 24:1329–1339. https://doi.org/10.1111/geb.12365

Van Looveren A, Klaise J (2019) Interpretable counterfactual explanations guided by prototypes. arXiv [csLG]

Wadoux A M J-C, Saby N P A, Martin M P (2023) Shapley values reveal the drivers of soil organic carbon stock prediction. SOIL 9:21–38. https://doi.org/10.5194/soil-9-21-2023

Zurell D, Elith J, Schröder B (2012) Predicting to new environments: tools for visualizing model behaviour and impacts on mapped distributions. Diversity & distributions 18:628–634. https://doi.org/10.1111/j.1472-4642.2012.00887.x

Zurell D, Franklin J, König C, et al (2020) A standard protocol for reporting species distribution models. Ecography 43:1261–1277. https://doi.org/10.1111/ecog.04960