Received: 19 December 2024 | Accepted: 17 May 2025

DOI: 10.1111/2041-210X.70074

Methods in Ecology and Evolution

PRACTICAL TOOLS

# *adm*: An R package for constructing abundance-based species distribution models

**Admir Cesar de Oliveira Junior**[1] | **Santiago José Elías Velazco**[2,3]

[1]Programa de Pós-Graduação em Biodiversidade Neotropical, Universidade Federal da Integração Latino-Americana, Foz do Iguaçu, Paraná, Brazil

[2]Department of Geography, San Diego State University, San Diego, California, USA

[3]Instituto de Biología Subtropical, Universidad Nacional de Misiones—CONICET, Puerto Iguazú, Misiones, Argentina

**Correspondence**
Santiago José Elías Velazco
Email: svelazco@sdsu.edu

**Funding information**
FONCyT, Agencia de Ciencia y Tecnología, Ministerio de Ciencia y Tecnología de Argentina, Grant/Award Number: PICTO-2022-10-00097

**Handling Editor:** Fernando González Taboada

## Abstract

1. Abundance-based species distribution models (ADM) correlate species abundance with environmental data to model and project abundance throughout space or time. This promising and still developing technique has gained significant attention in recent years.
2. Here, we present the *adm* R package developed to support the construction of ADM, including data preparation, model fitting, prediction and model exploration. This package offers several modelling approaches (i.e. algorithms) that can be fine-tuned and customized. Models can be predicted in geographic space and explored regarding performance and response curves. Because modelling workflows in *adm* are constructed based on a combination of distinct functions and simple outputs, *adm* can be easily integrated into other packages. To illustrate this, we constructed a full modelling procedure for the shrub species *Cynophalla retusa* using *adm*.
3. To date, *adm* provides 35 functions in three categories, (i) modelling: to tune, fit and validate models with nine different algorithms, with a suite of possible model-specific hyperparameters; (ii) post-modelling: to predict abundance across space and construct partial dependence plots to explore the relationships between abundance and environmental predictors; and (iii) miscellaneous tools: to support the workflow in all steps, including data handling, transformations and hyperparameter selection.
4. With *adm*, we intend to provide a flexible, straightforward and concise toolbox for ADM construction and expect it to help users develop and leverage the promising ADM field.

**KEYWORDS**
artificial neural networks, correlative models, model tuning, spatial ecology, species abundance models, species distribution models

## 1 | INTRODUCTION

Spatially explicit ecological models are an important approach in many research areas because they consider how spatially structured characteristics and constraints influence the modelled phenomena (DeAngelis & Yurek, 2017). One of the most widely used strategies for estimating species geographic distributions is the species distribution model (SDM, also known as ecological niche models or habitat suitability models). This technique uses occurrence data and environmental variables to model the environmental suitability or occurrence probability of a species and predict its distribution (Franklin, 2023). Despite the importance of SDM in ecology and conservation, other important ecological aspects, such as population density and trends, are often not modelled because of the lack of data (Hastings et al., 2020).

Abundance-based species distribution models (ADM) have recently been recognized as a distinct class of models (see Waldock et al., 2022); however, they were also commonly referred to as "SDM fitted with abundance data" or "Species Abundance Models" (e.g. García-Gómez et al., 2023; Kroetz et al., 2025). ADM are similar to SDM, as both are spatially explicit correlative models; however, they model and project abundance throughout space or time by correlating species abundance and environmental data (Anadón et al., 2010; Ehrlén & Morris, 2015; Yu et al., 2020). Predictions of species abundance are useful for assessing extinction risk, estimating the effects of climate and land-use change, understanding the environmental drivers of species abundance and performing conservation prioritization analysis (Villén-Pérez et al., 2020). Nonetheless, ADMs remain less developed than SDMs (Waldock et al., 2022), except for N-mixture models, which are appropriate for multiple visit abundance data. A few R packages support ADM fitting, tuning, evaluation and prediction, and most use only a few algorithms (Tables S1 and S2). Using multiple algorithms and performing hyperparameters tuning is crucial for diversifying species abundance modelling approaches and enabling selection of the best models or accounting for model uncertainty (Qiao et al., 2015; Thuiller et al., 2019).

Here, we introduce *adm*, a new R package designed to facilitate the development of ADM workflows. *adm* is one of the only packages offering fitting, tuning and model exploration of different modelling approaches, from generalized linear and additive models (GLM and GAM) (Rigby & Stasinopoulos, 2005) to different types of artificial neural networks (NET) implemented with *torch* (Falbel & Luraschi, 2024), which provide high architectural customization, native GPU (graphics processing unit) acceleration and more complex setups. Furthermore, *adm* is structured to support modelling workflows that can be easily integrated with other R packages.

## 2 | PACKAGE OVERVIEW

*adm* was inspired by the philosophy of the *flexsdm* R package, which enables users to build flexible modelling workflows by combining user-selected functions that return widely used R objects such as *terra* SpatRaster and *tidyverse* tibbles (Velazco et al., 2022). Furthermore, *adm* works as an independent extension of *flexsdm*, offering features adapted to abundance modelling. *adm* features can be used alone or integrated with *flexsdm* features, for example, various data partitioning approaches for model fitting and validation (k-fold, bootstrap or environmental and geographically structured partition), model calibration area delimitation, measuring model extrapolation and performing model truncation (Velazco et al., 2024). Building on the advances in developing conventional SDM, the integration of *adm* and *flexsdm* will improve the development of state-of-the-art ADM workflows.

Currently, *adm* provides 35 functions divided into three categories: modelling, post-modelling and miscellaneous tools (Figure 1). Modelling abundance is more challenging than modelling presence-absence because abundance can be measured in different ways (e.g. plot coverage percentage or absolute abundance). Therefore, *adm* is designed to handle different types of response variables by (i) offering different probability distributions for different algorithms, (ii) selecting suitable distributions based on response variable nature for GLM and GAM (*family_selector* function) or (iii) performing data transformations with different methods (*adm_transform* function). Additionally, *adm* does not create any new R object classes; rather, most function outputs are simple R lists (i.e. heterogeneous vectors that can contain different object classes, e.g. lists, tibbles, rasters and vectors). For instance, in the modelling functions, the outputs comprise an object of the original modelling framework and additional informative and useful tabular data. The simplicity of the *adm*'s output allows users to manipulate and explore results, making them more compatible with other packages.

## 3 | MODELLING FUNCTIONS

Selecting a modelling algorithm is a crucial step in developing a predictive model for species abundance and requires an understanding of the model assumptions and functionality. This can be a challenging task, and it is often recommended to test multiple algorithms (Qiao et al., 2015). *adm* facilitates modelling, hyperparameter tuning and validation of nine algorithms (Table S3), grouped into two function types, denoted by *fit_abund* and *tune_abund* prefixes. Both *fit_abund* and *tune_abund* validate models internally (see Table S3, full list of algorithms and hyperparameters).

*fit_abund* functions allow users to fit algorithms with default or user-specified hyperparameter values. However, tuning model hyperparameters influences algorithm performance and complexity and is therefore an important consideration when developing conventional SDMs (Fourcade, 2021). Choosing the optimal hyperparameter values for the data and modelling objective can significantly enhance model performance. Functions with the *tune_abund* prefix allow users to perform model tuning by using an array of model-specific hyperparameters. In *adm*, tuning is performed using a grid-search approach, that is, by evaluating model performance under an array of possible hyperparameter


**FIGURE 1** Overview of *adm* package functions structured in modelling (fitting and validation), post-modelling (model predictions and exploratory plots) and various utilities to support modelling workflows.

combinations. To implement this approach, the user provides a data frame with hyperparameters as columns and the hyperparameter values to be tested as rows (Appendix S1). Model tuning functions then iterate through the available hyperparameter values and select the best-performing model and its associated hyperparameter values. To reduce computational time, *tune_abund* functions support parallel processing built with *parallel* (R Core Team, 2024), *doParallel* (Microsoft & Weston, 2022a) and *foreach* (Microsoft & Weston, 2022b) packages. It is possible to evaluate models using one or more performance metrics during the model tuning process. For instance, use one metric for different performance metrics types. However, when using multiple performance metrics, their order is crucial because *model_selection* prioritises earlier metrics, sequentially selecting top-quartile models for each metric until a single model remains. Performance metrics for model evaluation are calculated using *adm_eval* (see below).

### 3.1 | Generalized linear and additive models

GLM and GAM are expansions of linear models and enable the modelling of non-linear predictor-response relationships, even when the data are not normally distributed (Hastie & Tibshirani, 1986; Nelder & Wedderburn, 1972). GLM assumes that the relationships between predictor and response variables are mediated by a link function that allows using different probability distributions (Nelder & Wedderburn, 1972). GAM uses a link and smoothing function, which captures non-linear relationships between response and predictor variables (Hastie & Tibshirani, 1986). In *adm*, GLM and GAM are based on the Generalized Additive Models for Location, Space and Shape framework (GAMLSS), using the *gamlss* package (Rigby & Stasinopoulos, 2005). GAMLSS offers >100 probability distributions that fit different types of response variables (e.g. between 0 and 1, discrete positive, zero-inflated; Figure S1) and the possibility of modelling any parameter that defines a family distribution (Stasinopoulos & Rigby, 2012). However, users should be aware that not every available distribution family is necessarily appropriate for modelling abundance. Therefore, the choice of distribution should be carefully considered and grounded in prior assumptions. In *adm*, users can fit GLM and GAM models using all the distribution families supported by *gamlss* (Figure S1). However, it is worth noting that the response variable must respect the family's assumptions; *tune_abund_glm and tune_abund_gam* automatically select the most suitable families if they are not provided within

the user-specified grid. GLM can be parameterized with different interaction orders between explanatory variables and the degree of polynomials. For GAM, users can control the smoothness degree used in a formula. For the GLM and GAM, parameters that define a distribution (i.e. sigma, nu and tau) can be modelled based on predictor variables.

### 3.2 | Generalized boosted regression models and extreme gradient boosting

Boosting algorithms are machine learning algorithms that sequentially train small models, each one improving upon the errors of the previous model, which is known as 'boosting' (Friedman et al., 2000; Friedman, 2001, 2002). In *adm*, boosting algorithms are supported in two different expansions and implementations of the original modelling framework (Friedman et al., 2000; Friedman, 2001, 2002), generalized boosted regression model (GBM) and extreme gradient boosting (XGB), via *gbm* and *xgboost* packages, respectively (Chen et al., 2024; Greg & G.B.M. Developers, 2024). Both are set to use trees as boosters but have significant differences in gradient computation, hyperparameters, overfitting prevention and regression tree construction (Chen & Guestrin, 2016) (Figures S2 and S3). The user can tune hyperparameters, such as tree depth and learning rate (Table S3).

### 3.3 | Random forest

Random forest (RAF) is a machine learning algorithm based on the ensemble of multiple decision trees trained with a bootstrapped version of the original dataset and predictors subset (Breiman, 2001). RAF has been widely used in ecology and distribution modelling, generally obtaining good performance, even with small datasets (Pichler & Hartig, 2023; Valavi et al., 2022). The algorithm was implemented via the *randomForest* package (Liaw & Wiener, 2002). Here, the user can set the number of trees grown in the forest and the number of predictors used for each decision tree (Table S3).

### 3.4 | Support vector machine

Support vector machine (SVM) is a machine learning algorithm that aims to define an optimal hyperplane determined by non-linear decision boundaries that split samples into different classes within a higher-dimensional space (Salcedo-Sanz et al., 2014). SVM was implemented using the *kernlab* package (Karatzoglou et al., 2004). SVM's *adm* function is set up to perform (epsilon) regressions, and the user can set the desired kernel, its parameters and the constraint violation cost (Table S3). SVM is adapted to work with Radial Basis and Laplacian kernels, but the user can experiment with different kernels and configurations; in which case, it is recommended to read the *kernlab* documentation.

### 3.5 | Artificial neural networks, deep neural networks and convolutional neural networks

Neural Networks are systems composed of interconnected neurons capable of learning complex non-linear data relationships. These neurons are organized into one or multiple layers called hidden layers (Alzubaidi et al., 2021). When a network features multiple serialized hidden layers, it is often called a deep neural network (DNN) (Alom et al., 2019). In this case, networks can be constructed with several architectures, combining multiple types of neurons, layers and functions (Pouyanfar et al., 2019). We refer to DNN as a fully connected, feedforward, backpropagation and artificial neuron network, which is the most common type of deep network (Alom et al., 2019). The neuron receives inputs, performs a weighted operation and feeds forward a value transformed by an activation function to the next layer (Schmidhuber, 2015). When a neural network features a single hidden layer, it can be called a Shallow Neural Network (Podder et al., 2021), referred to as NET herein and in *adm* documentation. These networks function similarly to DNN, but are less computationally intensive, although they perform well (Winkler & Le, 2017). Other common structures are the convolutional neural networks (CNN), which have filters or kernels that perform convolution operations across large multidimensional data matrices. This process sequentially generates "activation maps" between the layers, which allows the network to learn complex features from the data (Alzubaidi et al., 2021). Although there are no rules for their construction, excessively deep and large neural networks of any type tend to overfit (Pichler & Hartig, 2023). In ecology, these techniques have gained significant attention, encompassing a wide range of applications, including regressions and distribution models (Borowiec et al., 2022; Pichler & Hartig, 2023).

For DNN and CNN, *adm* uses the *torch* framework for R (Falbel & Luraschi, 2024). This allows the construction of highly customizable architectures, with the size and number of layers defined by the user. NET is based on the *nnet* package (Venables et al., 2002), which is a single-layer and less customizable option; however, it is much faster than DNN and CNN.

*adm* provides functions to help define the size and number of layers for constructing CNN and DNN. *generate_dnn_architecture* and *generate_cnn_architecture* functions help to easily construct neural networks. To facilitate the tuning process, the *generate_arch_list* function builds multiple architectures with different layer configurations. To systematically sample these architectures, *select_arch_list* can be used to reduce the list of architectures while maintaining a range of characteristics. In addition, users can manually construct a neural network using *torch* package syntax and use it within *fit_abund* and *tune_abund* functions.

### 3.6 | Model performance metrics *adm_eval*

Model evaluation metrics are calculated using the *adm_eval* function, which is implemented internally in the *fit_adm* and *tune_adm*

**TABLE 1** Model performance metrics, acronyms and their characteristics.

| Metric | Acronym | Range | Type |
|---|---|---|---|
| Spearman correlation | corr_spear | $[-1, 1]$ | Discrimination |
| Pearson correlation | corr_pear | $[-1, 1]$ | |
| Slope | slope | $(-\infty, +\infty)$ | |
| Intercept | inter | $(-\infty, +\infty)$ | |
| Mean absolute error | mae | $[0, +\infty)$ | Accuracy |
| Dispersion | pdisp | $[0, +\infty)$ | Precision |

functions. *adm_eval* returns a *tibble* with results for six performance metrics calculated between observed and predicted data (Table 1), based on Waldock et al. (2022): (i) Spearman's and (ii) Pearson's correlations, (iii) Mean Absolute Error, which consists of the absolute value of the average residual, (iv) Intercept and (v) Slope of a linear model fitted with observed abundance as a function of predicted abundance and (vi) Dispersion, calculated as the ratio between the standard deviation of predicted and observed abundance.

## 4 | POST-MODELLING FUNCTIONS

### 4.1 | Model prediction

In *adm*, spatial predictions for all algorithms are performed using *adm_predict*, using rasterized predictor variables as input. This function can simultaneously predict multiple models with prediction transformation, accounting for negative and scale-transformed values. Transforming negative values can be useful for algorithms that do not specify a statistical distribution (e.g. some machine learning approaches).

### 4.2 | Partial dependence plot and partial bivariate dependence plots

Partial dependence plots allow for the exploration of marginal response curves by linearly varying the values of one predictor while maintaining other constants. In *adm*, partial dependence plots and their bivariate version can easily be constructed with *p_abund_pdp* and *p_abund_bpdp*, which return a *ggplot2* object (Wickham, 2016). Both functions require only the output of *tune_* and *fit_* functions.

## 5 | MISCELLANEOUS TOOLS

### 5.1 | Dataset and variable manipulation

Models and predictions can be constructed by transforming response and predictor variables. To facilitate this process, *adm_transform* can scale predictor variables in rasters or response variables in a table. It can also return the values to the original scale if necessary. Because algorithms are sensitive to the number of zeros (Barbet-Massin et al., 2012; Liu et al., 2019), the *balance_dataset* can be used to perform absence data thinning by randomly selecting absences to equilibrate the number of presences and absences to a given ratio.

### 5.2 | Additional tools

Several other *adm* functionalities can be useful in the modelling workflow. *family_selector* identifies suitable distribution families to use in GAM and GLM based on the range and type of response variable (Figure S1). *model_selection* iterates over the performances dataset of a given model to select the best-performing hyperparameter combination based on user-defined performance metrics. *model_selection* is implemented in each *tune_abund_* function, but users can utilize it independently, for example, to reselect the best hyperparameter combination based on different metrics, without the need to tune the model again. *adm_summarize* concatenates performance tables from different models into one single table.

## 6 | EXAMPLE

We illustrate the use of *adm* and its integration with *flexsdm* (Velazco et al., 2022) by modelling the abundance of *Cynophalla retusa* (Griseb.) Cornejo & Iltis (Capparaceae) (Appendix S2; de Oliveira Junior & Velazco, 2025). It is a shrub native to northeastern Argentina, Paraguay, Bolivia and central Brazil, and is distributed mainly in dry biomes. We compiled and standardized data from the first (1998–2002) and second (2020) national forestry surveys in Argentina (MAyDS, 2022; SAyDS, 2005), constructing an abundance (individuals/ha) and absence dataset (sites with 0 individuals/ha). Using the *adm::balance_dataset*, we balanced presence (sites with >0 individuals/hectare) and absence at a 1:1 ratio. The absences were limited to the species training area, constructed as a 200-km buffered minimum convex polygon around presence points, using *flexsdm::calib_area*. We performed a principal component analysis using *flexsdm::correct_colinvar*, with 35 climatic and edaphic variables (Table S4), and selected the first seven principal components that represented >90% of cumulative variance as predictors (Table S5). To perform model validation accounting for models' spatial transferability (Roberts et al., 2017), we partitioned the dataset into three spatial blocks using *flexsdm::part_sblock*. To construct the models, we used DNN, RAF and GLM algorithms, fitted and validated by *adm::tune_abund_dnn*, *adm::tune_abund_raf* and *adm::tune_abund_glm*, aiming to maximize both Pearson's correlation and MAE (Table S6). Because DNN often performs better with scaled data (LeCun et al., 1998), input response data were standardized by Z-score using *adm::adm_transform* before fitting this algorithm and all architectures tested featured batch normalization between layers, generated with *adm::generate_arch_list*. Predictions were generated, restricted to the species calibration area, with *adm::adm_predict*.

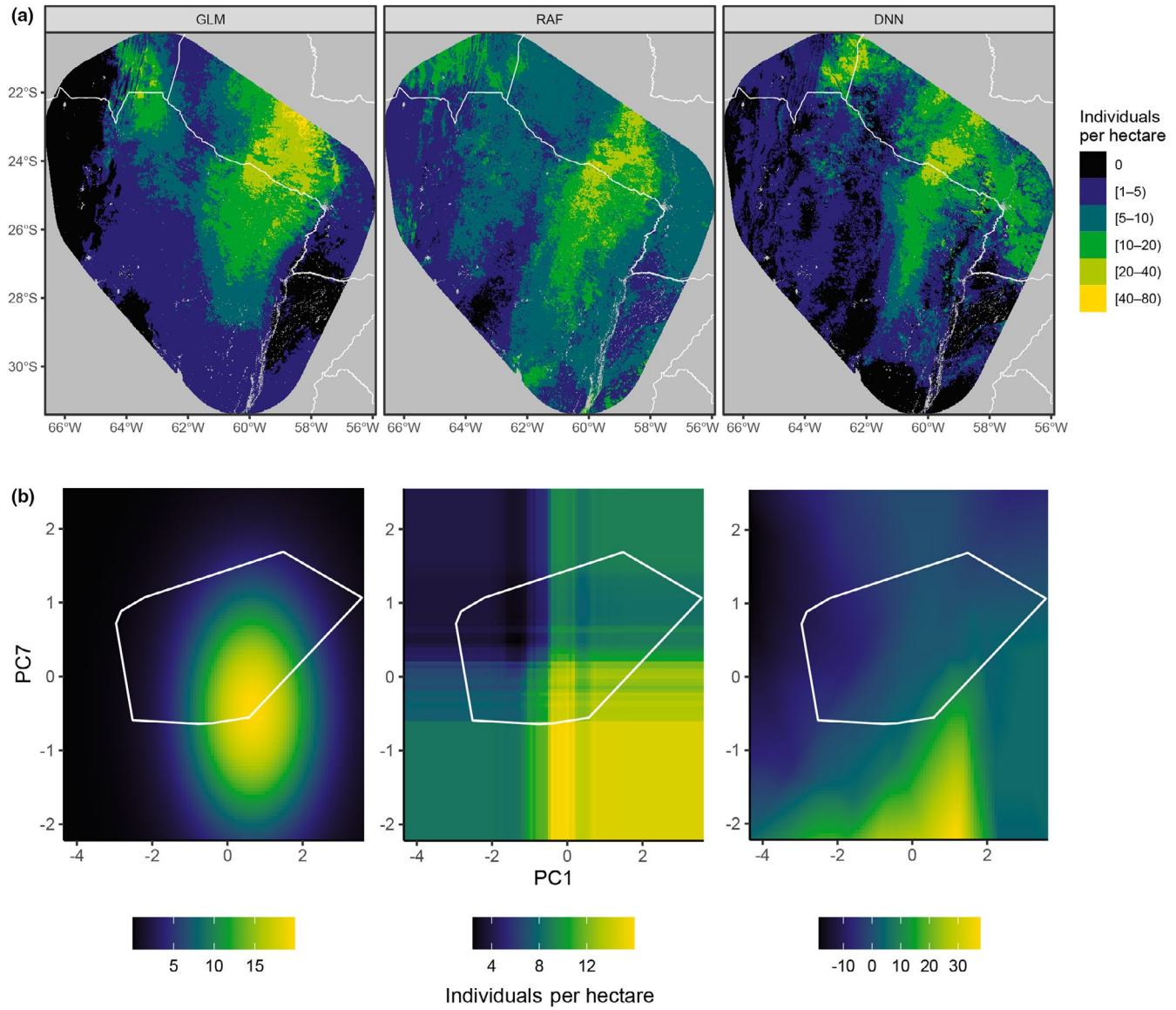


FIGURE 2 Abundance-based species distribution model for *Cynophalla retusa* predicted by generalized linear models (GLM), random forest (RAF) and deep neural networks (DNN). (a) Abundance maps predicted by each algorithm within training area. Predictions were classified into intervals to facilitate visualization. (b) Partial bivariate dependence plots for each model. White polygon in b represents the range of environmental conditions explored by abundance data.

To explore how models extrapolate, we produced bivariate partial dependence plots with *adm*::*p_abund_bpdp*, taking as an example the first and seventh principal components (Figure 2; Figures S4–S9 for all bivariate and univariate partial dependence plots). Comprehensive functions documentation and illustrative examples are available on the *adm* website (https://sjevelazco.github.io/adm/). The package is available on GitHub (https://github.com/sjevelazco/adm).

## 7 | CONCLUSION

The *adm* R package provides functions to construct a full workflow to model and predict species abundance in the geographic and environmental space. We highlight the possibility of using a variety of highly customizable algorithms and provide several functions for predicting and exploring ADMs. The complete integration of *adm* with *flexsdm* creates a holistic environment for modelling conventional species' presence-absence and species' abundance, allowing users to seamlessly combine and compare both approaches. In the future, we aim to expand *adm*'s features by implementing algorithm ensembles, ensembles of small models (Breiner et al., 2015), new algorithms, variable importance and other evaluation metrics. We expect that *adm* will help users to further develop the promising ADM field by providing a flexible, straightforward, integrated and concise toolbox.

### AUTHOR CONTRIBUTIONS

Admir Cesar de Oliveira Junior and Santiago José Elías Velazco: Conceptualization, methodology, formal analysis, software. Admir

Cesar de Oliveira Junior: Visualization, writing—original draft. Admir Cesar de Oliveira Junior and Santiago José Elías Velazco: writing—review and editing.

## ACKNOWLEDGEMENTS

We thank B. Rose and J. Franklin for reading and editing the first draft of the manuscript. ACOJ thanks Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES-DS) for the master's degree scholarship. SJEV thanks the Center for Open Geographical Science (COGS), Department of Geography, at San Diego State University for the research support and funding from FONCyT, Agencia de Ciencia y Tecnología, Ministerio de Ciencia y Tecnología de Argentina (PICTO-2022-10-00097).

## CONFLICT OF INTEREST STATEMENT

We declare no conflict of interest.

## PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.70074.

## DATA AVAILABILITY STATEMENT

Data and code available via https://doi.org/10.6084/m9.figshare.28688045 (de Oliveira Junior & Velazco, 2025).

## ORCID

*Admir Cesar de Oliveira Junior* https://orcid.org/0000-0002-1946-0475

*Santiago José Elías Velazco* https://orcid.org/0000-0002-7527-0967

## REFERENCES

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics*, *8*(3), 292. https://doi.org/10.3390/electronics8030292

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

Anadón, J. D., Giménez, A., & Ballestar, R. (2010). Linking local ecological knowledge and habitat modelling to predict absolute species abundance on large scales. *Biodiversity and Conservation*, *19*(5), 1443–1454. https://doi.org/10.1007/s10531-009-9774-4

Barbet-Massin, M., Jiguet, F., Albert, C. H., & Thuiller, W. (2012). Selecting pseudo-absences for species distribution models: How, where and how many? *Methods in Ecology and Evolution*, *3*(2), 327–338. https://doi.org/10.1111/j.2041-210X.2011.00172.x

Borowiec, M. L., Dikow, R. B., Frandsen, P. B., McKeeken, A., Valentini, G., & White, A. E. (2022). Deep learning as a tool for ecology and evolution. *Methods in Ecology and Evolution*, *13*(8), 1640–1660. https://doi.org/10.1111/2041-210X.13901

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Breiner, F. T., Guisan, A., Bergamini, A., & Nobis, M. P. (2015). Overcoming limitations of modelling rare species by using ensembles of small models. *Methods in Ecology and Evolution*, *6*(10), 1210–1218. https://doi.org/10.1111/2041-210X.12403

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., Li, Y., & Yuan, J. (2024). *xgboost: Extreme gradient boosting*. https://CRAN.R-project.org/package=xgboost

de Oliveira Junior, A. C., & Velazco, S. J. E. (2025). *Adm modeling example code and data*. 8404919 Bytes. https://doi.org/10.6084/M9.FIGSHARE.28688045

DeAngelis, D. L., & Yurek, S. (2017). Spatially explicit modeling in ecology: A review. *Ecosystems*, *20*(2), 284–300. https://doi.org/10.1007/s10021-016-0066-z

Ehrlén, J., & Morris, W. F. (2015). Predicting changes in the distribution and abundance of species under environmental change. *Ecology Letters*, *18*(3), 303–314. https://doi.org/10.1111/ele.12410

Falbel, D., & Luraschi, J. (2024). *torch: Tensors and neural networks with "GPU" acceleration*. https://torch.mlverse.org/docs

Fourcade, Y. (2021). Fine-tuning niche models matters in invasion ecology. A lesson from the land planarian *Obama nungara*. *Ecological Modelling*, *457*, 109686. https://doi.org/10.1016/j.ecolmodel.2021.109686

Franklin, J. (2023). Species distribution modelling supports the study of past, present and future biogeographies. *Journal of Biogeography*, *50*(9), 1533–1545. https://doi.org/10.1111/jbi.14617

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, *28*(2). https://doi.org/10.1214/aos/1016218223

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5). https://doi.org/10.1214/aos/1013203451

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

García-Gómez, R. E., Aceves-Medina, G., Villalobos, H., Jiménez Rosenberg, S. P. A., & Durazo, R. (2023). Predictive performance from abundance distribution models of *Vinciguerria lucetia* larvae in the southern portion of the California current system using XGBOOST. *Deep Sea Research Part II: Topical Studies in Oceanography*, *212*, 105336. https://doi.org/10.1016/j.dsr2.2023.105336

Greg, R., & G.B.M. Developers. (2024). *gbm: Generalized boosted regression models*. https://CRAN.R-project.org/package=gbm

Hastie, T., & Tibshirani, R. (1986). Generalized additive models. *Statistical Science*, *1*(3). https://doi.org/10.1214/ss/1177013604

Hastings, R. A., Rutterford, L. A., Freer, J. J., Collins, R. A., Simpson, S. D., & Genner, M. J. (2020). Climate change drives poleward increases and equatorward declines in marine species. *Current Biology*, *30*(8), 1572–1577. https://doi.org/10.1016/j.cub.2020.02.043

Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). Kernlab—An S4 package for kernel methods in R. *Journal of Statistical Software*, *11*(9). https://doi.org/10.18637/jss.v011.i09

Kroetz, A. M., Dedman, S., & Carlson, J. K. (2025). Predictive modeling of juvenile Smalltooth sawfish habitats: Challenges and opportunities for conservation. *Ecology and Evolution*, *15*(1), e70592. https://doi.org/10.1002/ece3.70592

LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. (1998). Efficient BackProp. In G. B. Orr & K.-R. Müller (Eds.), *Neural networks: Tricks of the trade* (pp. 9–50). Springer. https://doi.org/10.1007/3-540-49430-8_2

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22.

Liu, C., Newell, G., & White, M. (2019). The effect of sample size on the accuracy of species distribution models: Considering both presences and pseudo-absences or background sites. *Ecography*, *42*(3), 535–548. https://doi.org/10.1111/ecog.03188

MAyDS. (2022). *Segundo Inventario Nacional de Bosques Nativos: Informe Nacional*. Ministerio de Ambiente y Desarrollo Sostenible de la Nación.

Microsoft, & Weston, S. (2022a). *doParallel: Foreach parallel adaptor for the "parallel" package* [manual]. https://CRAN.R-project.org/package=doParallel

Microsoft, & Weston, S. (2022b). *foreach: Provides foreach looping construct*. https://CRAN.R-project.org/package=foreach

Nelder, J. A., & Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, *135*(3), 370. https://doi.org/10.2307/2344614

Pichler, M., & Hartig, F. (2023). Machine learning and deep learning—A review for ecologists. *Methods in Ecology and Evolution*, *14*(4), 994–1016. https://doi.org/10.1111/2041-210X.14061

Podder, P., Bharati, S., Mondal, M. R. H., Paul, P. K., & Kose, U. (2021). Artificial neural network for cybersecurity: A comprehensive review. *arXiv*. http://arxiv.org/abs/2107.01185

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., & Iyengar, S. S. (2019). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, *51*(5), 1–36. https://doi.org/10.1145/3234150

Qiao, H., Soberón, J., & Peterson, A. T. (2015). No silver bullets in correlative ecological niche modelling: Insights from testing among many potential algorithms for niche estimation. *Methods in Ecology and Evolution*, *6*(10), 1126–1136. https://doi.org/10.1111/2041-210X.12397

R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Rigby, R. A., & Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C: Applied Statistics*, *54*(3), 507–554. https://doi.org/10.1111/j.1467-9876.2005.00510.x

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J. J., Schröder, B., Thuiller, W., Warton, D. I., Wintle, B. A., Hartig, F., & Dormann, C. F. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, *40*(8), 913–929. https://doi.org/10.1111/ecog.02881

Salcedo-Sanz, S., Rojo-Álvarez, J. L., Martínez-Ramón, M., & Camps-Valls, G. (2014). Support vector machines in engineering: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4*(3), 234–267. https://doi.org/10.1002/widm.1125

SAyDS. (2005). *Primer Inventario Nacional de Bosques Nativos*. Ministerio de Salud y Ambiente de la Nación.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Stasinopoulos, M., & Rigby, R. (2012). *gamlss: Generalized additive models for location scale and shape (p. 5.4–22)* [dataset]. https://doi.org/10.32614/CRAN.package.gamlss

Thuiller, W., Guéguen, M., Renaud, J., Karger, D. N., & Zimmermann, N. E. (2019). Uncertainty in ensembles of global biodiversity scenarios. *Nature Communications*, *10*(1), 1446. https://doi.org/10.1038/s41467-019-09519-w

Valavi, R., Guillera-Arroita, G., Lahoz-Monfort, J. J., & Elith, J. (2022). Predictive performance of presence-only species distribution models: A benchmark study with reproducible code. *Ecological Monographs*, *92*(1), e01486. https://doi.org/10.1002/ecm.1486

Velazco, S. J. E., Rose, M. B., De Andrade, A. F. A., Minoli, I., & Franklin, J. (2022). flexsdm: An r package for supporting a comprehensive and flexible species distribution modelling workflow. *Methods in Ecology and Evolution*, *13*(8), 1661–1669. https://doi.org/10.1111/2041-210X.13874

Velazco, S. J. E., Rose, M. B., De Marco, P., Regan, H. M., & Franklin, J. (2024). How far can I extrapolate my species distribution model? Exploring shape, a novel method. *Ecography*, *2024*(3), e06992. https://doi.org/10.1111/ecog.06992

Venables, W. N., Ripley, B. D., & Venables, W. N. (2002). *Modern applied statistics with S* (4th ed.). Springer.

Villén-Pérez, S., Heikkinen, J., Salemaa, M., & Mäkipää, R. (2020). Global warming will affect the maximum potential abundance of boreal plant species. *Ecography*, *43*, 801–811. https://doi.org/10.1111/ecog.04720

Waldock, C., Stuart-Smith, R. D., Albouy, C., Cheung, W. W. L., Edgar, G. J., Mouillot, D., Tjiputra, J., & Pellissier, L. (2022). A quantitative review of abundance-based species distribution models. *Ecography*, *2022*(1), 05694. https://doi.org/10.1111/ecog.05694

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

Winkler, D. A., & Le, T. C. (2017). Performance of deep and shallow neural networks, the universal approximation theorem, activity cliffs, and QSAR. *Molecular Informatics*, *36*(1–2), 1600118. https://doi.org/10.1002/minf.201600118

Yu, H., Cooper, A. R., & Infante, D. M. (2020). Improving species distribution model predictive accuracy using species abundance: Application with boosted regression trees. *Ecological Modelling*, *432*, 109202. https://doi.org/10.1016/j.ecolmodel.2020.109202

## SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

**Figure S1.** Distribution families available in *gamlss* and implemented in *adm*.

**Figure S2.** Distributions available for GBM (*gbm*) (Greg & G.B.M. Developers, 2024).

**Figure S3.** Objectives available for XGB (*xgboost*) (Chen et al., 2024).

**Figure S4.** Bivariate partial dependence plots of a generalized linear model used for modeling *Cynophalla retusa* abundance.

**Figure S5.** Univariate dependence plots of a generalized linear model used for modeling *Cynophalla retusa* abundance.

**Figure S6.** Bivariate partial dependence plots of a random forest used to model *Cynophalla retusa* abundance.

**Figure S7.** Univariate partial dependence plots of a random forest a used to model *Cynophalla retusa* abundance.

**Figure S8.** Bivariate partial dependence plots of a deep neural network used for modeling *Cynophalla retusa* abundance.

**Figure S9.** Univariate partial dependence plots of a deep neural network used for modeling *Cynophalla retusa* abundance.

**Table S1.** R packages that construct spatially explicit abundance-based species models compared to *adm*.

**Table S2.** Features of *adm* compared with other R packages.

**Table S3.** Algorithms implemented in *adm* and their respective *fit_* and *tune_* functions, hyperparameters, and source packages.

**Table S4.** Source and names of variables used to perform the principal component analysis.

**Table S5.** Variance explained by principal components selected from the principal component analysis performed with climate and edaphic data.

**Table S6.** Performance metrics for random forest (RAF), deep neural network (DNN), and generalized linear models (GLM).

**Appendix S1.** Creating the grid of parameters to be tested in *tune_abund* functions.

**Appendix S2.** Data sent and code to reproduce *Cynophalla retusa* example.

**How to cite this article:** de Oliveira Junior, A. C., & Velazco, S. J. E. (2025). *adm*: An R package for constructing abundance-based species distribution models. *Methods in Ecology and Evolution*, *00*, 1–9. https://doi.org/10.1111/2041-210X.70074