Model-R: A Framework for Scalable and Reproducible Ecological Niche Modeling

Andrea Sánchez-Tapia[1], Marinez Ferreira de Siqueira[1], Rafael Oliveira Lima[1], Felipe Sodré M. Barros[2], Guilherme M. Gall[3], Luiz M. R. Gadelha Jr.[3], Luís Alexandre E. da Silva[1], and Carla Osthoff[3]

[1] Botanic Garden of Rio de Janeiro, Rio de Janeiro, Brazil
{andreasancheztapia, marinez, rafael, estevao}@jbrj.gov.br
[2] International Institute for Sustainability, Rio de Janeiro, Brazil
f.barros@iis-rio.org
[3] National Laboratory for Scientific Computing, Petrópolis, Brazil
{gmgall, lgadelha, osthoff}@lncc.br

Abstract. Spatial analysis tools and synthesis of results are key to identifying the best solutions in biodiversity conservation. The importance of process automation is associated with increased efficiency and performance both in the data pre-processing phase and in the post-analysis of the results generated by the packages and modeling programs. The Model-R framework was developed with the main objective of unifying pre-existing ecological niche modeling tools into a common framework and building a web interface that automates steps of the modeling process and occurrence data retrieval. The web interface includes RJabot, a functionality that allows for searching and retrieving occurrence data from *Jabot*, the main reference on botanical collections management system in Brazil. It returns data in a suitable format to be consumed by other components of the framework. Currently, the tools are multi-projection, they can thus be applied to different sets of temporal and spatial data. Model-R is also multi-algorithm, with seven algorithms available for modeling: BIOCLIM, Mahalanobis distance, Maxent, GLM, RandomForest, SVM, and DOMAIN. The algorithms as well as the entire modeling process may be parametrized using command-line tools or through the web interface. We hope that the use of this application, not only by modeling specialists but also as a tool for policy makers, will be a significant contribution to the continuous development of biodiversity conservation analysis. The Model-R web interface can be installed locally or on a server. A software container is provided to automate the installation.

Keywords: species distribution modeling, ecological niche modeling, science gateways, scalability, provenance

1 Introduction

Ecological Niche Modeling (ENM) has been widely used for over a decade [1] [2] [3] [4]. In recent years ENM approaches have become an essential tool for species

conservation, ecology and evolution studies, as well for systematic conservation and restoration planning [5]. These models use species occurrence data and predictor variables that are combined to form statistical and theoretical models resulting in projections in the geographic space that represent the potential geographic distribution of a species [6]. The environmental suitability maps [7], generated by the models inform how similar a particular area is to the area where the species occurs, thus identifying the potential area for occupation by the species, from the predictor variables selected.

Ecological niche modeling comprises several stages, which require knowledge of many concepts and techniques related to various fields of biology, such as biodiversity, biogeography, as well as climate and data processing tools, before, during and after obtaining the model [8][5]. The biotic data processing step consists of obtaining, evaluating and preparing the points of presence and, in some cases, of absence of the species to be modeled. In this process, it is fundamental to perform data cleaning with the removal of inaccurate or unreliable data. In the step of treatment and choice of environmental layers, one obtains and selects the layers to be used in the analysis. Traditionally, it is necessary to use a Geographic Information System (GIS) tools for clipping and adjusting the resolution and cropping the raster layers to the modeling extension, requiring a reasonable knowledge of the tool. This task can be even more time-consuming when dealing with a large dataset. The use of specific data types by the algorithms, and their various forms of parametrization, requires a reasonable knowledge of programming for their full use. The importance of process automation is associated with increased efficiency and performance both in the data pre-processing phase and in the post-analysis of the results generated by the packages and modeling programs, which is the main objective of this work. The elimination of external tools for data acquisition and preparation, as well as their standardization, reduces the possibility of errors, confers reproducibility and improves the speed of the modeling process, making the whole process more efficient.

The modeling process consists of many steps, as described in [8], some of which consume considerable time to be performed by traditional means. A resource available for tackling this problem is the R statistical environment, which features various possibilities of automation but does require some knowledge of programming for obtaining the desired outcomes in this process. The main objective of this work was to package modeling procedures as R functions and to create an application (Model-R) that allows, either via command-line or through a web interface, to perform ecological niche modeling, overcoming the most common barriers and providing approaches for data entry steps, data cleaning, choice of predictor variables, parametrization of algorithms, and post-analysis as well as the retrieval of the results. A list of acronyms and variable definitions is presented in Table 1.

2 Model-R Framework

The Model-R framework for ecological niche modeling is given by a set of ecological niche modeling functions (`dismo.mod`, `final.models`, `ensemble`), functions for retrieving species occurrences records (Rjabot and Rgbif) and the graphic user interface. It allows researchers to use their own data. The framework is divided in front and backend; some functions are presented at web interface that abstracts and automates the main steps involved in the ecological niche modeling process. This is a dynamic process, and our goal is that this interface will evolve and incorporate more and more aspects of the framework. All these components were implemented in R and are described in the subsections that follow.

2.1 Frontend

The main focus of the web application for Model-R is the development of an interface for the modeling process, allowing users without programming knowledge to perform the steps of the modeling process consistently, avoiding the concern with script coding and concentrating on the data and its processing workflow. To do so, we adapted the modeling functions into a Shiny application [9]. The Shiny package [9] is a web application framework for R, allowing the creation of interactive applications that can be accessed from devices with internet access, such as computers, tablets, and mobile phones. Thus, the application provides a graphical interface where users can easily choose biotic and abiotic data, perform the data cleaning on occurrence records, choose algorithms and their parameters. They can also download the results, as well as the script of the modeling process that allow its execution without the use of the Model-R web application. The use of the script as a stand-alone application allows for more precise adjustment of the parameters or adjustments that were not possible in the web interface. To make the application and process user-friendly, we separated the features by steps, following the modeling process described in [8].

The following steps of the modeling process are available in the application: biotic data entry; data cleaning; choice of abiotic variables; cutting off the geographic extension; choosing the algorithm and its parameters; visualization of results; and downloading the resulting data.

Biotic data entry. This stage represents the entry of biotic data in the system. A modeling project can be created using the "Create Project" feature, this allows for keeping track of the modeling experiments performed. Creating a project allows one to assign a name and thus organize and store the information generated. Biotic data can be given as input to the application in three ways: queries to the GBIF database, queries to the *Jabot* database, and uploading CSV files. CSV files allow for uploading occurrence records not present in GBIF and *Jabot* from other databases after conversion to this format. The RJabot package makes the query to the *Jabot* database (Figure 1). These records are given by species name, latitude, and longitude. At the end of the biotic data entry step, a map with the occurrence records is displayed. In July 2017, GBIF contained approximately 10 million species occurrence records about Brazil, 70% of which

were published by its Brazilian node, the Brazilian Biodiversity Information System (SiBBr) [10].

Fig. 1. Output of *getOccurrence* showing occurrence points obtained from Jabot.

Data cleaning. This step allows cleaning the biotic data entered into the application. It has two features: "Eliminate duplicate data" and "Delete Occurrence point". "Eliminate Duplicate Data" removes occurrence records entered to the application that have the same value for latitude and longitude. "Delete occurrence point" eliminates points that were evaluated by the user as erroneous in their location and will not be used in modeling. Using the interface, the user clicks the button "Delete duplicate" or selects the point it wants to eliminate suspicious data. After that, the user can also save the final biotic dataset, after the data cleaning process.

Abiotic data entry. This step is responsible for the entry of abiotic variables and definition of the geographic extensions of the modeling process and its projection. The first step is to set the spatial extension of modeling process (i.e.: the extent to which the modeling will be done, also understood as study area). The extensions can be defined directly on the map, which displays the occurrence points selected in the previous steps. Regarding spatial projection, the application allows users to define a different extent to project ENM in another region. This can be useful, for instance, for checking the ability of a species to become invasive in the given region. Also, it is possible to define a projection in time, in instance, to the past (Pleistocene/Holocene) or future (2050 and 2070) using

Worldclim dataset[4] and Bio-ORACLE variables [11]. Independently of the spatial and temporal projection chosen, the user might define the spatial resolution (i.e. pixel size) of the abiotic dataset. For development, we used the resolution of 10 arc minutes (for Worldclim variables) and 9.2 km (for Bio-ORACLE variables), due to storage space and processing speed reasons. The main database technologies that optimize storage and speed processing are already under study, so the application supports others resolutions, like 30 seconds, 2.5, 5 arc minutes.

The map, the occurrence points, and the geographic extensions are displayed using the Leaflet [12]. The package allows for zooming and interacting with the map. The application is configured to work with Wordclim and Bio-ORACLE to retrieve abiotic data and allow for other variables to be added manually to the application.

Once the abiotic variables are defined, the Model-R application displays the variables considering the extension defined by the user, and a table with charts containing the correlation values between them (see Figure 2, step 4), allowing to verify the correlated variables. Strongly correlated variables can impair the prediction performance and statistics of the modeling process [13] [14].

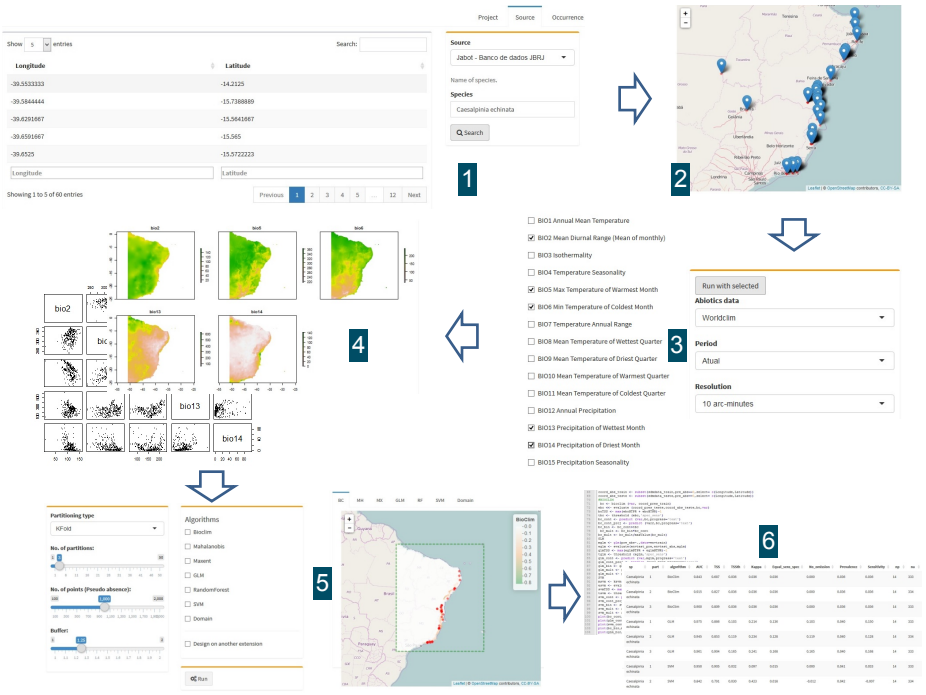

Fig. 2. Modeling steps in the web interface of Model-R.

[4] http://www.worldclim.org

3 Modeling process and backend

The next step in the web application, modeling process, is the core of the species distribution modeling workflow and was implemented as a three-step procedure, wrapped in R functions, called `dismo.mod()` (in reference to the dismo package [15] from which it draws the main structure and functions), `final.models()` and `ensemble()`.

`dismo.mod()` takes the input data, partitions it by cross-validation, fits the models for each partition and writes the evaluation statistics tables, using function `evaluate ()` in the dismo package, with some modifications, such as the calculation of TSS for each partition. It writes the raw models, i.e. the continuous outputs, in raster and image formats. Writing to the hard disk allows keeping the main memory uncluttered. The structure of the function draws both on the dismo [15] and the biomod2 [16] tutorials.

`final.model()` joins the fitted models for each partition into a final model per species per algorithm. It can select the best partitions according to their TSS or AUC value. The default is selecting by $TSS > 0.7$, but this can be changed by the user. The function also allows choosing which algorithms will be processed. Otherwise, it will read all algorithms available from the statistics table, and to use a mask to crop the models to a subset of the fitting geographic area. Finally, it cuts the continuous models by the threshold that maximizes the TSS of the model and averages these models.

`ensemble()` computes the average of the final models, to obtain an ensemble model per species and retaining only the algorithms and partitions that were selected previously. It can also focus on the areas where the algorithms exhibited consensus. The default is 0.5, which corresponds to a Weighted Majority Rule Ensemble to reduce variability between the algorithms in final models so that the final models only retains areas predicted by at least half of the algorithms [17].

The application interface runs this framework in the background, but the user can adjust the following parameters:

Partition Number. The number of times the model will be generated for each selected algorithm and, consequently, the number of times the k-fold partitioning will be performed. (dividing the total data set in k mutually exclusive subsets of the same size, using $k-1$ for parameter estimation and algorithm training and the remaining subset to evaluate the accuracy of the model).

Number of pseudo-absences. Number of points sampled randomly in the background for use in the modeling process.

Modeling algorithms Seven algorithms are available: BIOCLIM, Mahalanobis, Maxent, DOMAIN available in the dismo package [15] GLM ('stats'), RandomForest ('randomForest') and SVM ('kernlab'). BIOCLIM, Malahanobis, and DOMAIN are based on simple statistical techniques, such as environmental distance. GLM is based on regression techniques. Lastly, Maxent, RandomForest, and SVM are based on machine learning techniques.

Buffer should be applied during the sampling of pseudo-absences. This is an inclusive buffer, it calculates the distance between the occurrence points and use the maximum or the mean geographic distance between the occurrences of the species within which pseudo-absences will be generated.

Project on another extension. The application reprojected the model to different extensions (spatial or temporal) from the modeling process obtained on the creation extension.

At the end of the execution, k continuous models, k binary models and one ensemble model are generated for each species and algorithm, as displayed in Figure 2 (step 5). The values obtained from the validation process are stored as a table, and their values are presented in Figure 2 (step 6). A brief description of each variable is presented in Table 1.

Table 1. Description of variables generated by the modeling process.

Variable name	**Description**
Sp	Species name
Part	Partition number
Algorithm	Modeling algorithm employed
AUC	Computed Area Under Curve
TSS	True skill statistic = (sensitivity + specifity) - 1
Kappa	Cohen's Kappa coefficient
No Omission	Threshold where there is no omission
Prevalence	Prevalence
Sensitivity	Sensitivity
TSSth	Threshold = (sens+esp)
Np	Number of presences
Na	Number of Absences

4 Reproducibility

Provenance information [18] is given by the documentation of the conception and execution of computational processes, including the activities that were executed and the respective data sets consumed and produced by them. Applications of provenance include reproducibility of computational processes, sharing and reuse of knowledge, data quality evaluation and attribution of scientific results [19]. Reproducibility is one of the important features of Model-R. The inclusion of this feature is motivated by many academic journals recommending that authors of computational studies should also provide the required data sets, tools, and workflows used to generate the results [20] [21] so that reviewers and readers could better validate them. For each modeling project specified and executed in Model-R, the following information is available for download: the R script,

illustrated in Figure 2 (step 6) that allows for reproducing the steps that were performed to produce results of the modeling process and to re-execute the modeling process without using the web interface of Model-R; a CSV file containing the resulting variables from the modeling process; the occurrence records used after data cleaning; the raster files in the GeoTIFF format generated by the application; a raster file in the GeoTIFF format with an ensemble of the models generated; raster files in the GeoTIFF format with the projection of the model into another geographic extension. These are only generated when the "Project into another extension" option is selected.

5 Case Study and Evaluation

A case study was performed with woody plants of the Brazilian Atlantic Forest and is described next.

Species occurrence data. The original plant names database (3,952 plant names and 171,144 original records) were compiled from SpeciesLink[5] and NeoTropTree[6] (List of species with number of records – appendix 1) and corrected according to the Catalog of Plants and Fungi of Brazil (CPFB)[7], using R package flora [22], which is based on the List's IPT database. The CPFB publishes the official List of the Brazilian Flora, meeting Target 1 of the Global Strategy for Plant Conservation. The catalog recognized and checked 3,910 names. The 42 names that were not found by the LSBF were looked for in The Plant List[8] (TPL) and then in the Taxonomic Name Resolution Service[9] (TNRS), as implemented in the R packages Taxonstand [23] and taxize [24]. [25]. The information from the CPFB, TPL, and TNRS was cross-checked, and when there were conflicts, the names from the CPFB were given priority. For each species the complete occurrence data was treated for (1) records that fell out of the Brazilian limit, (2) duplicated records, (3) non-duplicated records that fell in the same 1km-pixel. Only species with at least 100 unique occurrences (deleting duplicated within each pixel) were maintained, and of these, to overcome bias of with marginal occurrence for Atlantic Rainforest, only species with more than 50% of occurrences in the Atlantic Rain Forest were considered. After all these procedures, a sub-sample of the 96 species (35,672 presence records) that presented the largest numbers of samples was chosen to compose the given woody plants case study (Figure 3, left).

Environmental data. As environmental predictors, 28 variables with spatial resolution of $1km^2$ were compiled and organized. Those variables were summarized by PCA axes, from which the first ten axes (about 95% of the data variation) were used to run models. Aspect variable was edited and had its sin and cosin created to be used as variables.

[5] http://splink.cria.org.br/
[6] http://prof.icb.ufmg.br/treeatlan/
[7] http://floradobrasil.jbrj.gov.br/
[8] http://www.theplantlist.org/
[9] http://tnrs.iplantcollaborative.org/

Environmental Niche Modeling. Environmental niche models were built for each species, using dismo.mod, final.model, and ensemble functions.

A three-fold cross validation procedure was performed. Random pseudo-absence points ($nback = 2 \times n$) were sorted within a maximum distance buffer (the radius of the buffer is the maximal geographic distance between the occurrence points) and divided into three groups, for training and testing purposes.

For each partition ($k = 3$) and algorithm, a model was built, and its performance was tested by calculating the True Skill Statistic [26]. The authors found that TSS scores were largely unaffected by prevalence and values from 0.6 to 1.0 were considered as a good adjustment of the model accuracy. Because of that, only models with $TSS > 0.7$ were retained. Selected partitions were cut by the threshold that maximizes their TSS, and the resulting binary models were averaged to generate a model per algorithm. The scale in these final models is equivalent to the number of partitions that predict the species presence (it goes from 0 to $\frac{n}{n}$ in $\frac{1}{n}$ intervals where n is the number of selected models). The ensemble model (e.g. joining models from different algorithms) was obtained by averaging the final models for each algorithm. A species potential richness map was generated by summing the binary final models, cut by the average threshold that maximizes TSS values (Figure 3, right).

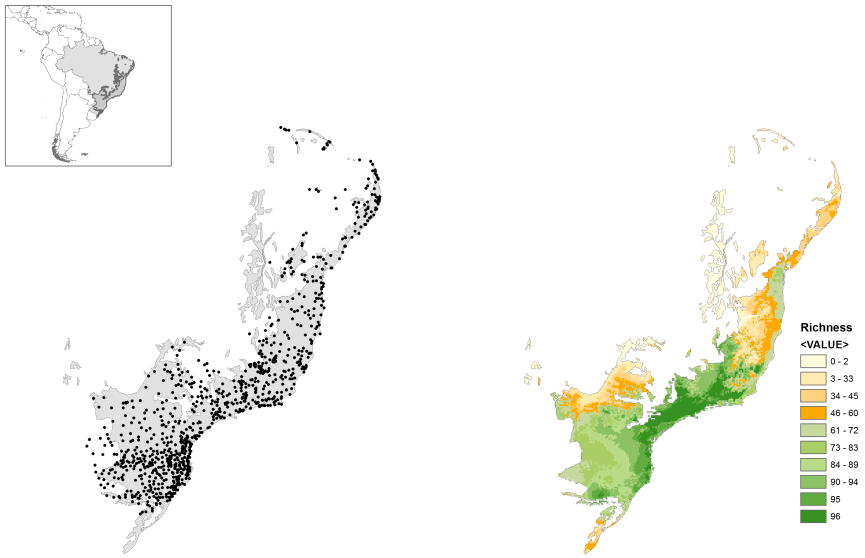

Fig. 3. Map with original occurrence records (left) and richness map generated by analyzing Model-R output data (right).

Performance and Parallelization The `dismo.mod()` function, in which the modeling process of Model-R is based, was entirely sequential in its first version. Models for all species of interest were generated one after another. To improve performance, parallel processing was employed. Now, if n cores are available, models for n species can be generated simultaneously. The snowfall [27] R package provided support for the parallelization. It provides functions for parallel collection processing. `sfLapply`, for instance, is the parallel version of the standard `lapply`, which applies some function to every element of an array, producing a new array with the results.

The effects of the parallelization on performance can be seen in Figure 4. Each point in the plot is the arithmetic mean of the time elapsed to do three executions of `dismo.mod()`. Models for 96 species were generated, varying the number of cores from 1 to 64. The algorithms used were RandomForest, SVM, and Maxent. The variability in execution time for 96 species can be explained in part by the parallelization strategy used, i.e. one thread per species. The total time that it takes to apply all the modeling algorithms can be significantly different from one species to another.

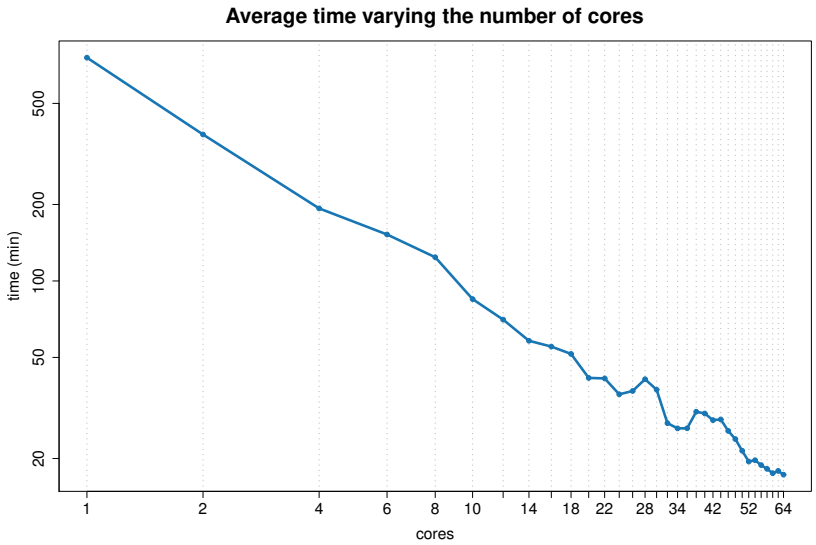

Fig. 4. Parallelization effects on performance.

The creation of separate functions for each of the modeling algorithms that `dismo.mod()` can fit was another important optimization. In its first version, all algorithms were generating models in the context of a single function. The memory allocated to the variables used by one algorithm was never released even if the referenced algorithm had finished its work. R is a programming language

with garbage collector [28] meaning it releases memory when an object is no longer used. It does this by counting how many references point to each object and when the counter reaches zero, removes that object. Since `dismo.mod()` was keeping at least one reference to the variables used by all selected algorithms for all the runtime of the function, a lot of memory was being occupied unnecessarily. The separation did not make the modeling process faster but allowed the generation of more models per node because of the smaller memory footprint. The generation of models for a single species was performed using approximately 5GB of resident memory. Resident memory is a metric that gets closer to the actual memory budget of a process [29]. The version with separate functions for each modeling algorithm uses half of this memory.

6 Related Work

As parameters for comparison, two related services in this area were considered: the *Biomodelos* portal [30], developed by the Humboldt Institute in Colombia, and the Virtual Biodiversity e-Laboratory (BioVel) [31], an initiative supported by the European Union. These two examples were chosen because they represent two distinct efforts from the standpoint of the internal and external target audience of the system.

BioVel provides, via a web interface, a service that allows management of scientific workflows [32] for biodiversity. Several pre-defined activities can be composed to form these workflows, as an example, the following features were developed and are available in BioVel: geographic and temporal selection of occurrences; data cleaning; Taxonomic name resolution; modeling algorithms ecological niches (openModeller) [33]. Such activities can be composed freely in complex scientific workflows for performing various analyses on biodiversity. This flexibility of service and the range of applications available in its catalog, generate a plurality of results provided by the service that can be difficult to assess regarding quality and suitability, since the service is freely accessible and does not have a methodology systematic qualification of models where experts can criticize, comment and change the generated results.

The Biomodelos portal [30] is intended for species distribution modeling, which is carried out and published on the website by the Humboldt Institute modeling team. The most interesting feature of biomodelos, absent from similar portals is the existence of a network of taxonomists, who are also users of the portal, which evaluates each species distribution published on the website. The taxonomy experts have access to metadata about how the species distribution models were executed and can assign a note to the generated model, add notes to them or geographically edit the distribution map (excluding, for example, records in areas with known absences). Thus, species distributions published in Biomodelos are accompanied by information to support the decision maker in assessing their quality and fitness for the use.

Other initiatives based on R include SDM [34] and Wallace [35], and based on scientific workflow management system include Kepler [36], VisTrails [37] [38]

or in the cloud computing environment [39] [40] [41]. They have some similarity with our application as well as some striking differences, especially in terms of functionality, such as the lack of scalability in the implementation of the models and the absence of provenance recording.

7 Conclusion

In this work, we presented Model-R, a framework for species distribution modeling. It abstracts away cumbersome steps usually involved in this type of modeling, such as data acquisition and cleaning, by providing a productive web interface that allows for customizing key steps of the process, including the pre-processing of biotic and abiotic data and the post-analysis of the results. The RJabot package, for instance, allows for easy retrieval of species occurrence records from the Rio de Janeiro Botanical Garden Herbarium (RB) [42], one of the most complete sources of information about the Brazilian flora. The scalable execution of the modeling process is enabled through the use of parallel programming libraries available for the R environment. Having separate functions per algorithm also presents an opportunity for further exploration of parallelism. Currently only parallelism by species is used. All models for a given species are generated by the same core even if more than one algorithm is used. Parallelism by algorithm is feasible as well. Model-R also enables reproducibility of the modeling process by providing the data sets generated and scripts in R that allow for reproducing the steps used to generate them. The application supports applying the modeling process to different sets of temporal or spatial data. Maxent, RandomForest, and all the algorithms supported by the dismo package are supported by Model-R, and their parameters can be customized through its web interface. We expect the application to become a valuable tool for scientist working with analysis and synthesis of biodiversity data, and for decision-makers in biodiversity conservation.

As future work, we plan to better automate the generation of raster files containing abiotic data by using GIS tools, such as PostGIS. These are currently generated manually for some pre-defined resolutions and copied to the Model-R application server. We also plan to further improve the scalability of the application by adapt it to run on petascale computing resources of the Brazilian National System for High-Performance Computing[10] [43] using the Swift [44] parallel scripting system, which gathers provenance information [45] [46]. Additionally, we are working on porting the modeling scripts to Big Data platforms. In particular, we are adapting them to the Spark platform [47] using its R interface [48].

Model-R is available as open-source software on Github[11]. To facilitate its installation, we also built a software container that is available on Docker Hub[12]. This software container is synchronized to the Github repository, i.e. any update

[10] `http://sdumont.lncc.br`
[11] `https://github.com/Model-R/Model-R`
[12] `https://hub.docker.com/r/modelr/shinyapp/`

to the source code on Github triggers the production of an updated software container.

Acknowledgments

This work has been supported by CNPq, SiBBr, and FAPERJ.

The original publication will be available at www.springerlink.com:
Sánchez-Tapia, A., de Siqueira, M., Lima, R., Barros, F., Gall, G., Gadelha, L., and da Silva, L. (2017). Model-R: A Framework for Scalable and Reproducible Ecological Niche Modeling. High Performance Computing - Fourth Latin American Conference, CARLA 2017. Communications in Computer and Information Science, vol. 796. Springer, 2017.
`http://link.springer.com/chapter/10.1007/978-3-319-73353-1_15`

References

1. Araújo, M.B., Williams, P.H.: Selecting areas for species persistence using occurrence data. Biological Conservation **96**(3) (dec 2000) 331–345
2. Engler, R., Guisan, A., Rechsteiner, L.: An improved approach for predicting the distribution of rare and endangered species from occurrence and pseudo-absence data. Journal of Applied Ecology **41**(2) (apr 2004) 263–274
3. Ortega-Huerta, M.A., Peterson, A.T.: Modelling spatial patterns of biodiversity for conservation prioritization in North-eastern Mexico. Diversity and Distributions **10**(1) (jan 2004) 39–54
4. Chen, Y.: Conservation biogeography of the snake family Colubridae of China. North-Western Journal of Zoology **5**(2) (2009) 251–262
5. Peterson, A.T., Soberón, J., Pearson, R.G., Anderson, R.P., Martínez-Meyer, E., Nakamura, M., Araújo, M.B.: Ecological Niches and Geographic Distributions. Princeton University Press (2011)
6. Anderson, R.P., Lew, D., Peterson, A.: Evaluating predictive models of species' distributions: criteria for selecting optimal models. Ecological Modelling **162**(3) (apr 2003) 211–232
7. Sillero, N.: What does ecological modelling model? A proposed classification of ecological niche models based on their underlying methods. Ecological Modelling **222**(8) (apr 2011) 1343–1346
8. Santana, F., de Siqueira, M., Saraiva, A., Correa, P.: A reference business process for ecological niche modelling. Ecological Informatics **3**(1) (jan 2008) 75–86
9. Chang, W.: shiny: Web Application Framework for R. https://cran.r-project.org/web/packages/shiny (2016)
10. Gadelha, L., Guimarães, P., Moura, A.M., Drucker, D.P., Dalcin, E., Gall, G., Tavares, J., Palazzi, D., Poltosi, M., Porto, F., Moura, F., Leo, W.V.: SiBBr: Uma Infraestrutura para Coleta, Integração e Análise de Dados sobre a Biodiversidade Brasileira. In: VIII Brazilian e-Science Workshop (BRESCI 2014). Proc. XXXIV Congress of the Brazilian Computer Society. (2014)
11. Tyberghein, L., Verbruggen, H., Pauly, K., Troupin, C., Mineur, F., De Clerck, O.: Bio-ORACLE: a global environmental dataset for marine species distribution modelling. Global Ecology and Biogeography (2012)

12. Agafonkin, V.: Leaflet - a JavaScript library for interactive maps. http://leafletjs.com/ (2016)
13. Guisan, A., Zimmermann, N.E.: Predictive habitat distribution models in ecology. Ecological Modelling **135**(2-3) (dec 2000) 147–186
14. Lomba, A., Pellissier, L., Randin, C., Vicente, J., Moreira, F., Honrado, J., Guisan, A.: Overcoming the rare species modelling paradox: A novel hierarchical framework applied to an Iberian endemic plant. Biological Conservation **143**(11) (nov 2010) 2647–2657
15. Hijmans, R.J., Elith, J.: dismo: Species Distribution Modeling. https://cran.r-project.org/web/packages/dismo (2016)
16. Thuiller, W., Lafourcade, B., Engler, R., Araújo, M.B.: BIOMOD - A platform for ensemble forecasting of species distributions. Ecography **32**(3) (2009) 369–373
17. Araújo, M.B., Whittaker, R.J., Ladle, R.J., Erhard, M.: Reducing uncertainty in projections of extinction risk from climate change: Uncertainty in Species' Range Shift Projections. Global Ecology and Biogeography **14**(6) (jun 2005) 529–538
18. Freire, J., Koop, D., Santos, E., Silva, C.: Provenance for Computational Tasks: A Survey. Computing in Science & Engineering **10**(3) (may 2008) 11–21
19. Gadelha, L., Mattoso, M.: Applying Provenance to Protect Attribution in Distributed Computational Scientific Experiments. In Ludäscher, B., Plale, B., eds.: Provenance and Annotation of Data and Processes. Volume 8628 of Lecture Notes in Computer Science. Springer (2015) 139–151
20. Sandve, G.K., Nekrutenko, A., Taylor, J., Hovig, E.: Ten Simple Rules for Reproducible Computational Research. PLoS Computational Biology **9**(10) (oct 2013) e1003285
21. Wilson, G., Aruliah, D.A., Brown, C.T., Chue Hong, N.P., Davis, M., Guy, R.T., Haddock, S.H.D., Huff, K.D., Mitchell, I.M., Plumbley, M.D., Waugh, B., White, E.P., Wilson, P.: Best practices for scientific computing. PLoS biology **12**(1) (jan 2014) e1001745
22. Carvalho, G.: flora: Tools for Interacting with the Brazilian Flora 2020. https://cran.r-project.org/web/packages/flora/index.html (2016)
23. Cayuela, L., Oksanen, J.: Taxonstand: Taxonomic Standardization of Plant Species Names. https://cran.r-project.org/web/packages/Taxonstand (2016)
24. Chamberlain, S.A., Szöcs, E.: taxize: taxonomic search and retrieval in R. F1000Research **2** (jan 2013) 191
25. Chamberlain, S., Szoecs, E., Foster, Z., Boettiger, C., Ram, K., Bartomeus, I., Baumgartner, J., O'Donnell, J.: taxize: Taxonomic Information from Around the Web. https://cran.r-project.org/web/packages/taxize (2016)
26. Allouche, O., Tsoar, A., Kadmon, R.: Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (TSS). Journal of Applied Ecology **43**(6) (sep 2006) 1223–1232
27. Knaus, J.: snowfall: Easier cluster computing (based on snow). https://cran.r-project.org/web/packages/snowfall (2016)
28. Wickham, H.: Advanced R. Chapman and Hall/CRC (2014)
29. Simmonds, C.: Mastering Embedded Linux Programming. Packt (2015)
30. Biomodelos: Instituto Alexander von Humboldt. http://biomodelos.humboldt.org.co (2016)
31. Vicario, S., Hardisty, A., Haitas, N.: BioVeL: Biodiversity Virtual e-Laboratory. EMBnet.journal **17**(2) (sep 2011) 5
32. Liu, J., Pacitti, E., Valduriez, P., Mattoso, M.: A Survey of Data-Intensive Scientific Workflow Management. Journal of Grid Computing **13**(4) (mar 2015) 457–493

33. Souza Muñoz, M.E., Giovanni, R., Siqueira, M.F., Sutton, T., Brewer, P., Pereira, R.S., Canhos, D.A.L., Canhos, V.P.: openModeller: a generic approach to species' potential distribution modelling. GeoInformatica **15**(1) (aug 2009) 111–135
34. Naimi, B., Araújo, M.B.: sdm: a reproducible and extensible R platform for species distribution modelling. Ecography **39**(4) (feb 2016) 368–375
35. Kass, J., Anderson, R.P., Aiello-Lammens, M., Muscarella, B., Vilela, B.: Wallace (beta v0.1): Harnessing Digital Biodiversity Data for Predictive Modeling, Fueled by R. http://devpost.com/software/wallace-beta-v0-1-harnessing-digital-biodiversity-data-for-predictive-modeling-fueled-by-r (2016)
36. Pennington, D.D., Higgins, D., Peterson, A.T., Jones, M.B., Ludäscher, B., Bowers, S.: Ecological Niche Modeling Using the Kepler Workflow System. In Taylor, I.J., Deelman, E., Gannon, D.B., Shields, M., eds.: Workflows for e-Science. Springer London (jan 2007) 91–108
37. Talbert, C., Talbert, M., Morisette, J., Koop, D.: Data Management Challenges in Species Distribution Modeling. IEEE Bulletin of the Technical Committee on Data Engineering **36**(4) (2013) 31–40
38. Morisette, J.T., Jarnevich, C.S., Holcombe, T.R., Talbert, C.B., Ignizio, D., Talbert, M.K., Silva, C., Koop, D., Swanson, A., Young, N.E.: VisTrails SAHM: visualization and workflow management for species habitat modeling. Ecography **36**(2) (feb 2013) 129–135
39. Candela, L., Castelli, D., Coro, G., Pagano, P., Sinibaldi, F.: Species distribution modeling in the cloud. Concurrency and Computation: Practice and Experience **28**(4) (jul 2016) 1056–1079
40. Candela, L., Castelli, D., Coro, G., Lelii, L., Mangiacrapa, F., Marioli, V., Pagano, P.: An Infrastructure-oriented Approach for supporting Biodiversity Research. Ecological Informatics (aug 2014)
41. Amaral, R., Badia, R.M., Blanquer, I., Braga-Neto, R., Candela, L., Castelli, D., Flann, C., De Giovanni, R., Gray, W.A., Jones, A., Lezzi, D., Pagano, P., Perez-Canhos, V., Quevedo, F., Rafanell, R., Rebello, V., Sousa-Baena, M.S., Torres, E.: Supporting biodiversity studies with the EUBrazilOpenBio Hybrid Data Infrastructure. Concurrency and Computation: Practice and Experience **27**(2) (2015) 376–394
42. Forzza, R., Mynssen, C., Tamaio, N.; Barros, C.; Franco, L., Pereira, M.: As coleções do herbário. 200 anos do Jardim Botânico do Rio de Janeiro. Jardim Botânico do Rio de Janeiro, Rio de Janeiro (2008)
43. Mondelli, M.L., Galheigo, M., Medeiros, V., Bastos, B.F., Gomes, A.T.A., Vasconcelos, A.T.R., Gadelha Jr., L.M.R.: Integrating Scientific Workflows with Scientific Gateways: A Bioinformatics Experiment in the Brazilian National High-Performance Computing Network. In: X Brazilian e-Science Workshop. Anais do XXXVI Congresso da Sociedade Brasileira de Computação, SBC (2016) 277–284
44. Wilde, M., Hategan, M., Wozniak, J.M., Clifford, B., Katz, D.S., Foster, I.: Swift: A language for distributed parallel scripting. Parallel Computing **37**(9) (sep 2011) 633–652
45. Gadelha, L.M.R., Wilde, M., Mattoso, M., Foster, I.: Exploring provenance in high performance scientific computing. In: Proc. of the 1st Annual Workshop on High Performance Computing meets Databases - HPCDB '11, ACM Press (nov 2011) 17–20
46. Mondelli, M.L., de Souza, M.T., Ocaña, K., de Vasconcelos, A.T.R., Gadelha Jr, L.M.R.: HPSW-Prof: A Provenance-Based Framework for Profiling High Performance Scientific Workflows. In: Proc. of Satellite Events of the 31st Brazilian Symposium on Databases (SBBD 2016), SBC (2016) 117–122

47. Armbrust, M., Das, T., Davidson, A., Ghodsi, A., Or, A., Rosen, J., Stoica, I., Wendell, P., Xin, R., Zaharia, M.: Scaling Spark in the Real World: Performance and Usability. Proceedings of the VLDB Endowment **8**(12) (2015) 1840–1843
48. Venkataraman, S., Stoica, I., Zaharia, M., Yang, Z., Liu, D., Liang, E., Falaki, H., Meng, X., Xin, R., Ghodsi, A., Franklin, M.: SparkR: Scaling R Programs with Spark. In: Proceedings of the 2016 International Conference on Management of Data - SIGMOD '16, New York, New York, USA, ACM Press (2016) 1099–1104