Title: Shifting from closed-source graphical-interface to open-source programming environment: a brief tutorial on running Maxent in R

Authors: Xiao Feng[1*], Cassondra Walker[2], and Fikirte Gebresenbet[2]

[1]Institute of Environment, University of Arizona, Tucson, Arizona 85721-0137, USA

[2]Department of Integrative Biology, Oklahoma State University, Stillwater, Oklahoma 74078, USA

*Corresponding author: Xiao Feng, Institute of Environment, University of Arizona, Tucson, Arizona 85721-0137, USA, xiao.feng@okstate.edu

Abstract: Ecological niche modeling (ENM) is increasingly being used in studying the relationship between species distributions and environmental conditions. The development of ENM software/algorithms is heading toward open-source programming, for the advantage of efficiency in handling big data and incorporating new methods. Maxent is one of the commonly used ENM algorithms, but there has been limited information and efforts in implementing Maxent in an open-source programming environment (e.g., R). Therefore, we aim to fill the gap of knowledge for using Maxent in R. More specifically, we demonstrate the general implementation of Maxent in R based on a commonly used ENM procedure, provide a function that bridges the Maxent algorithm and R computing environment for easier use, and demonstrate the manipulation of a few crucial Maxent parameters in R. We expect our efforts will promote a shift of the Maxent user community from a graphical-interface to open-source programming.

Keywords: Maxent, R, open-source, clamping, ecological niche modeling, species distribution modeling

Introduction

The use of ecological niche modeling (ENM; or species distribution modeling) to quantify the relationship between species' presences and environmental conditions (Peterson *et al.*, 2011) has increased with the rapid development of GIS techniques and the accumulation and digitization of museum specimens. In the field of ENM, many modeling algorithms have been developed [e.g., GARP (Stockwell, 1999), ENFA (Hirzel *et al.*, 2002), and Maxent (Phillips *et al.*, 2004)]. Recently there is a trend toward development of open-source algorithms and software (e.g., Maxent; Phillips *et al.*, 2017), particularly in a programming environment (i.e., R; Naimi & Araújo, 2016; Thuiller *et al.*, 2016; Hijmans *et al.*, 2017; R Development Core Team, 2017).

The use of open-source algorithms in programing environments is particularly important and useful in the era of big data, when volumes and varieties of scientific data are accumulated at high speed. The input for ENM are mainly species' occurrence data and environmental data, which are both proliferating with advancements in technology and sampling efforts. The increase in species' occurrence data is due in large part to the digitization of museum specimens (Page *et al.*, 2015; Rogers, 2016) and citizen science projects (Sullivan *et al.*, 2009; Jackson *et al.*, 2015). This accumulation of species' locational data is most evident with the increase in records on the Global Biodiversity Information Facility (GBIF; http://www.gbif.org) database which has reached 0.79 billion species occurrences (accessed 25 July 2017). The other major data input for ENM are environmental data derived from geospatial techniques, which have increased largely from advances in Geographic Information Systems and remote sensing (Turner *et al.*, 2003; Wang *et al.*, 2010) and the creation of practical data products from them (Waltari *et al.*, 2014). With these advances, traditional software platforms (e.g., using graphical user interface by clicking) will be inadequate to handle such large amounts of data, whereas in a programing environment a simple loop function can efficiently process a task thousands of times based on an established procedure (e.g., Merow, 2017), with the further advantage of repeatability and accuracy. In addition to the increased amounts of available data, the development of new methods and tools is progressing rapidly (Muscarella *et al.*, 2014; Qiao *et al.*, 2015; Feng *et al.*, 2017b; Kass *et al.*, 2017). Traditional scientific software development will be insufficient in keeping pace with newly developed scientific methods, because of the limited effort and insight contained in a closed developing environment compared to a collective and open source developing environment (e.g. Github.com).

In the field of ENM, Maxent (Elith *et al.*, 2006; Phillips & Dudik, 2008) is probably the most used algorithm; likely because it is a presence only technique that is able to take advantage of the large amounts of occurrence data available and its user-friendly graphical-interface (Joppa *et al.*, 2013). The popularity of Maxent is evident with just a quick search of literature as its publications have been cited collectively thousands of times (accessed 25 July 2017). Maxent was originally a closed-source software and has recently been adapted to be open-source (Phillips *et al.*, 2017), though we believe that the graphical user interface persists as the popular method of operation.

There has been an effort to advocate the use of Maxent in R by Hijmans *et al.* (2017), but detailed demonstration of the how is lacking, creating a gap for novice users. The other gap of knowledge is that Maxent implements 67 parameters (Phillips, 2006), which are easy to manipulate in the graphical-interface but are not straight forward to implement in R. R users may sift through the scattered pieces of information on the Internet (e.g. Maxent-Google Groups, R mailing lists, or programmer communities), but no comprehensive and detailed documentation explaining or demonstrating implementation of Maxent parameters in R exists so far.

Additionally, ENM scholars found that three settings in Maxent may greatly influence model performance: features, regularization parameters, and clamping (Phillips & Dudik, 2008; Moreno-Amat *et al.*, 2015), thus manipulation of these parameters is important in model comparison and evaluation. Muscarella *et al.* (2014) insightfully provided a function that automatically implements models using different combinations of features and regularization parameters, but it is also useful to allow the user to explore different regularization values for different features (Phillips & Dudik, 2008).

Inspired by this lack of information, we identified three objectives which we expect to be important in promoting a shift of the Maxent user community from graphical-interface to open-source programming: 1) demonstrate the general implementation of Maxent in R based on a commonly used ENM procedure, 2) provide a function that bridges the Maxent algorithm and R computing environment for easier use, and 3) demonstrate the manipulation of features, regularization parameters, and clamping in R and explain potential tips and caveats of such implementations in R.

Summary of our case study

We used the nine-banded armadillo (*Dasypus novemcinctus*) as the study species and illustrated the general use of Maxent in R via R-Markdown document based on *dismo*, *raster*, and *knitr* packages (Xie, 2014; Hijmans *et al.*, 2016; Hijmans *et al.*, 2017). We provide detailed guidelines for loading GIS layers, downloading occurrence data, preparing data input for Maxent, and training and evaluating a Maxent model. We provided detailed annotations of the R scripts in the R-Markdown document (Appendix S1).

In addition to a well-documented guide, we have provided a function (Appendix S2) that bridges the Maxent parameters and R computing environment for easier use. This function allows R users to manipulate 26 of Maxent parameters that are commonly used in the Maxent interface but are not straight forward to manipulate in R. We provided a list of the parameters and their applications in Appendix S3.

Case study analysis

Package Preparation

Our demonstration of using Maxent in R relies on the following R packages: *dismo* (Hijmans *et al.*, 2017), *raster* (Hijmans *et al.*, 2016), *rgeos* (Bivand & Rundel, 2017), *rJava* (Urbanek, 2016), and *knitr* (Xie, 2014). Before running Maxent in R, the Maxent executable file (i.e. Maxent.jar) must be stored in a location that is accessible to the *dismo* package in R. We provided a line of code that downloads "Maxent.jar" and stores the file in the appropriate folder (i.e. the folder of dismo package; see Thread 4 in Appendix 1). Depending on the size of data processed in Maxent, users may exceed the memory limit of Java Virtual Machine; to address this issue, the function *options* can allocate the desired amount of memory for Java Virtual Machine (Thread 4 in Appendix 1).

Occurrence Data Preparation

We downloaded occurrence data of the nine-banded armadillo from GBIF using the function *gbif* and used additional functions to further filter and refine the downloaded dataset. Specifically, we eliminated records with invalid or duplicated coordinates using the functions *is.na* and *duplicated* (Thread 7 in Appendix 1). We then transformed the occurrence dataset into a spatial dataframe based on the associated coordinates using the *coordinates* function (Thread 8 in Appendix 1). We further removed records with longitudes greater than -40° or less than -110° that seemed erroneous based on ecological knowledge of the study species. The use of a loop function at this point could potentially, allow users to process occurrence data for a list of focal species.

Training Data Preparation

In our example, we used bioclimatic variables (downloaded from http://www.worldclim.org) as predictor variables for the ENM. We saved all environmental layers in one folder and loaded the desired files (i.e., files with *.bil* extension) through regular expression (Thread 5 in Appendix 1). Instead of using a world map or whole continent, we used a relatively smaller area for model training (i.e., buffers around occurrences; Thread 11 in Appendix 1). We restricted the bioclimatic layers to the study area using the functions *crop* and *mask* (Thread 12 in Appendix 1). Applying the *crop* function before *mask* is expected to save computation time by only masking a smaller extent.

We selected 10,000 random background points from the study area using the function *sampleRandom*. To make this random sampling reproducible, we specified a fixed seed for the *sampleRandom* function (Thread 13 in Appendix 1). Similarly, we randomly split the occurrence data into testing and training using a fixed seed.

Maxent can use either spatial or tabular data as inputs. The spatial data refer to spatial points (i.e., geographic coordinates) and raster layers of predictors, whereas tabular data pertain to presences and background records with associated environmental conditions. We employ tabular data in our example because it allows for more flexibility in model implementation, such as background data manipulation (e.g., Barbet-Massin *et al.*, 2012). Tabular data require formatting the environmental conditions for both occurrences and background points (Thread 15 in Appendix 1), and requires a binary vector of 1s and 0s that correspond to records of presence and background points (Thread 16 in Appendix 1).

Model Training, Projecting, and Evaluation

While constructing our model, we specified the output folder using parameter *path*; however, more parameters of Maxent can potentially be implemented via the parameter *args*. To facilitate the application of Maxent parameters, we designed a function called *prepPara* that allows for straight forward implementation of Maxent parameters in R. In particular, users can specify a desired combination of features, such as linear and quadratic, designate a beta-multiplier for different features, and enable or disable the clamp module of Maxent (Appendix 2).

We demonstrated two methods for model projection. The first method is similar to the procedure used by the Maxent interface, where the model projection is completed immediately after model training, by specifying the projection layers/data (Thread 23 in Appendix 1). The second method is to complete the projection post hoc using the function *predict* after obtaining a Maxent model (Thread 18 in Appendix 1). A Maxent model can be projected on a set of GIS layers or a dataframe that includes the same predictors used in model training.

After obtaining a Maxent model, the *evaluate* function can calculate the training AUC value if training occurrences and background points are provided, or the testing AUC value given testing occurrences (Thread 19 in Appendix 1). Threshold dependent evaluation indices can also be calculated if the threshold parameter *tr* is provided (Thread 20 in Appendix 1).

Discussion

Maxent is undoubtedly a robust modeling algorithm that will continue to attract research interests in invasive species distribution, climate change, conservation, and epidemic outbreaks (Reed *et al.*, 2008; Mainali *et al.*, 2015; Feng *et al.*, 2017a). We demonstrated the general usage of Maxent in R, provided a function that bridges the Maxent parameters and R computing environment, and exemplified the manipulation of a few crucial Maxent parameters in R. We hope our demonstration of using Maxent in R will help users learn necessary skills, as well as strengthen the efficiency of future research in the field of ENM. We hope that bridging Maxent and R will lower the threshold of methodology explorations for the Maxent community. In the era of big data, scientific software that is single client-based, closed-source graphical-interface will likely be replaced by working environments featured by open-source and programming, or potentially be replaced by server-based/cloud-based web application (Qiao *et al.*, 2012; Feng *et al.*, 2017b).

Supporting Information

Appendix S1. A brief tutorial on runing Maxent in R; available at https://github.com/shandongfx/workshop_maxent_R/blob/master/code/Appendix1_case_study.md

Appendix S2. Source code of a function that prepares Maxent parameters; available at https://github.com/shandongfx/workshop_maxent_R/blob/master/code/Appendix2_prepPara.R

Appendix S3. Implementation of Maxent parameters; available at https://github.com/shandongfx/workshop_maxent_R/blob/master/code/Appendix3_maxentParameters_v2.pdf

References

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**, 327-338.

Bivand, R. & Rundel, C. (2017) *rgeos: interface to Geometry Engine - Open Source (GEOS)*. R package version 0.3-23. Retrieved from https://cran.r-project.org/package=rgeos.

Elith, J., Graham, C.H., Anderson, R.P., et al. (2006) Novel methods improve prediction of species' distributions from occurrence data. *Ecography*, **29**, 129-151.

Feng, X., Castro, M.C., McBee, K. & Papeş, M. (2017a) Hiding in a cool climatic niche in the Tropics? an assessment of the ecological biogeography of hairy long-nosed armadillos (*Dasypus pilosus*). *Tropical Conservation Science*, **10**, 1-13.

Feng, X., Castro, M., Linde, E. & Papeş, M. (2017b) Armadillo Mapper: a case study of an online application to update potential distribution estimates. *Tropical Conservation Science*, **10**, 1-5.

Hijmans, R.J., Phillips, S., Leathwick, J. & Elith, J. (2017) *dismo: species distribution modeling*. R package version 1.1-4. Retrieved from http://cran.r-project.org/package=dismo.

Hijmans, R.J., Etten, J.v., Cheng, J., Mattiuzzi, M., Sumner, M., Greenberg, J.A., Lamigueiro, O.P., Bevan, A., Racine, E.B. & Shortridge, A. (2016) *raster: geographic data analysis and modeling*. R Package version 2.5-8. Retrieved from http://cran.r-project.org/web/packages/raster/.

Hirzel, A.H., Hausser, J., Chessel, D. & Perrin, N. (2002) Ecological-niche factor analysis: how to compute habitat-suitability maps without absence data? *Ecology*, **83**, 2027-2036.

Jackson, M.M., Gergel, S.E. & Martin, K. (2015) Citizen science and field survey observations provide comparable results for mapping Vancouver Island White-tailed Ptarmigan (*Lagopus leucura saxatilis*) distributions. *Biological Conservation*, **181**, 162-172.

Joppa, L.N., McInerny, G., Harper, R., Salido, L., Takeda, K., O'Hara, K., Gavaghan, D. & Emmott, S. (2013) Troubling trends in scientific software use. *Science*, **340**, 814-815.

Kass, J.M., Vilela, B., Aeillo-Lammens, M.E., Muscarella, R. & Anderson, R.P. (2017) *Wallace: A modular platform for reproducible ecological modeling. Version 0.6.3.* Available at: (accessed

Mainali, K.P., Warren, D.L., Dhileepan, K., McConnachie, A., Strathie, L., Hassan, G., Karki, D., Shrestha, B.B. & Parmesan, C. (2015) Projecting future expansion of invasive species: comparing and improving methodologies for species distribution modeling. *Global Change Biology*, **21**, 4464-4480.

Merow, C. (2017) *BIEN range methods description*. Available at: http://bien.nceas.ucsb.edu/bien/wp-content/uploads/2017/06/BIEN3RangeMethodsSummary.pdf (accessed September 19 2017).

Moreno-Amat, E., Mateo, R.G., Nieto-Lugilde, D., Morueta-Holme, N., Svenning, J.-C. & García-Amorena, I. (2015) Impact of model complexity on cross-temporal transferability in MaxEnt species distribution models: an assessment using paleobotanical data. *Ecological Modelling*, **312**, 308-317.

Muscarella, R., Galante, P.J., Soley-Guardia, M., Boria, R.A., Kass, J.M., Uriarte, M. & Anderson, R.P. (2014) ENMeval: An R package for conducting spatially independent evaluations and estimating optimal model complexity for Maxent ecological niche models. *Methods in Ecology and Evolution*, **5**, 1198-1205.

Naimi, B. & Araújo, M.B. (2016) sdm: a reproducible and extensible R platform for species distribution modelling. *Ecography*, **39**, 368-375.

Page, L.M., MacFadden, B.J., Fortes, J.A., Soltis, P.S. & Riccardi, G. (2015) Digitization of biodiversity collections reveals biggest data on biodiversity. *BioScience*, **65**, 841-842.

Peterson, A.T., Soberón, J., Pearson, R.G., Anderson, R.P., Martínez-Meyer, E., Nakamura, M. & Araújo, M.B. (2011) *Ecological Niches and Geographic Distributions*. Princeton University Press, Princeton, NJ.

Phillips, S.J. (2006) *A brief tutorial on Maxent*. Available at: http://www.cs.princeton.edu/~schapire/maxent/tutorial/tutorial.doc (accessed 1 August 2012).

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3346v1 | CC BY 4.0 Open Access | rec: 13 Oct 2017, publ: 13 Oct 2017

Phillips, S.J. & Dudik, M. (2008) Modeling of species distributions with Maxent: new extensions and a comprehensive evaluation. *Ecography*, **31**, 161-175.

Phillips, S.J., Dudík, M. & Schapire, R.E. (2004) A maximum entropy approach to species distribution modeling. In: *Proceedings of the Twenty-First International Conference on Machine Learning*, pp. 655–662. ACM press, Banff, Alberta, Canada.

Phillips, S.J., Anderson, R.P., Dudík, M., Schapire, R.E. & Blair, M.E. (2017) Opening the black box: an open-source release of Maxent. *Ecography*, doi: 10.1111/ecog.03049.

Qiao, H., Lin, C., Ji, L. & Jiang, Z. (2012) mMWeb - an online platform for employing multiple ecological niche modeling algorithms. *PLoS ONE*, **7**, e43327.

Qiao, H., Lin, C., Jiang, Z. & Ji, L. (2015) Marble algorithm: a solution to estimating ecological niches from presence-only records. *Scientific Reports*, **5**, 14232-14232.

R Development Core Team (2017) *R: a language and environment for statistical computing*. Retrieved from http://www.r-project.org/.

Reed, K.D., Meece, J.K., Archer, J.R. & Peterson, A.T. (2008) Ecologic niche modeling of *Blastomyces dermatitidis* in Wisconsin. *PLoS ONE*, **3**, e2034.

Rogers, N. (2016) Museum drawers go digital. *Science*, **352**, 762-765.

Stockwell, D. (1999) The GARP modelling system: problems and solutions to automated spatial prediction. *International Journal of Geographical Information Science*, **13**, 143-158.

Sullivan, B.L., Wood, C.L., Iliff, M.J., Bonney, R.E., Fink, D. & Kelling, S. (2009) eBird: a citizen-based bird observation network in the biological sciences. *Biological Conservation*, **142**, 2282-2292.

Thuiller, W., Georges, D., Engler, R. & Breiner, F. (2016) *biomod2: ensemble platform for species distribution modeling*. R package version 3.3-7. Retrieved from https://cran.r-project.org/package=biomod2.

Turner, W., Spector, S., Gardiner, N., Fladeland, M., Sterling, E. & Steininger, M. (2003) Remote sensing for biodiversity science and conservation. *Trends in Ecology & Evolution*, **18**, 306-314.

Urbanek, S. (2016) *rJava: low-level R to Java interface*. R package version 0.9-8. Retrieved from https://cran.r-project.org/package=rJava.

Waltari, E., Schroeder, R., McDonald, K., Anderson, R.P. & Carnaval, A. (2014) Bioclimatic variables derived from remote sensing: assessment and application for species distribution modelling. *Methods in Ecology and Evolution*, **5**, 1033-1042.

Wang, K., Franklin, S.E., Guo, X. & Cattet, M. (2010) Remote sensing of ecology, biodiversity and conservation: a review from the perspective of remote sensing specialists. *Sensors*, **10**, 9647-9667.

Xie, Y. (2014) knitr: a comprehensive tool for reproducible research in R. *Implementing Reproducible Computational Research* (ed. by V. Stodden, F. Leisch and R.D. Peng), p. 448. CRC Press, Chapman and Hall.