Journal of Biogeography

SUPPORTING INFORMATION

Making better MAXENT models of species distributions: complexity, overfitting, and evaluation

Aleksandar Radosavljevic and Robert P. Anderson

Appendix S1: Procedure for masking environmental variables.

For the masked geographically structured approach, we took background environmental data only from the areas of the bins used in model calibration (Anderson & Raza, 2010). To do so, we 'masked' out the area corresponding to the bin being withheld for evaluation by using a dummy variable. The dummy variable had a value of zero in the regions corresponding to the calibration bins and no data for the region corresponding to the bin of localities with which the model was to be evaluated. When the resulting model was projected to the full study region to allow for evaluation, the dummy variable was assigned a value of zero for the full study region.

Appendix S2: Instructions for running MAXENT from a command shell using .bat files.

Some implementations of MAXENT, such as the tuning experiments described here, require the generation of a large number of models. While it is possible to do this from the user interface, running the models from the command line greatly expedites the process. The command-line arguments can be run individually from a command shell or combined into a batch file and run in sequence by calling the batch file a single time. The latter allows the user to automate some aspects of the tuning process, greatly reducing the amount of time required to generate a large number of models. All of the settings available through the user interface can be adjusted via the command line.

Below is an example of a few lines from a batch file like those used to generate one of the many models used in this study:

```
java -mx512m -jar maxent.jar -e C:\GIScovs\Hetgrant\GeoPart\A togglelayerselected=alt -s
C:\GIS\Hetgrant\GeoPart\Locs\Cal_A.csv
-j C:\GIScovs\Hetgrant\GeoPart\Proj -b 0.25 -o C:\GIS\Hetgrant\GeoPart\Results\A\0.25 -K -
P nowarnings notooltips -a
```

Here, we parse the code above into constituent elements and explain them. The first portion of the commands above (java -mx512m -jar maxent.jar) calls java, tells it to allocate 512 MB of memory to this run, and to launch MAXENT. The flags following this tell MAXENT: where to find the environmental data (e) and to exclude altitude, where the sample file/s is/are located, where the projection environmental data (j) are located, what value of regularization multiplier (b) to use, and where to save the output (o). In addition, we have turned on picture making (K; maps of the predictions) and response curves (P); suppressed warning messages (nowarnings) and tooltips (notooltips); and turned on the autorun function, so that each model runs immediately after the preceding one without requiring the user to bypass warnings and press 'enter' or 'run' for each iteration. These constitute just a few examples of the flags available to the user; a complete list appears in the MAXENT help file.

To run models in batch mode, we recommend that the user write the commands in a simple text editor and then save the finished product as a batch file with the extension .bat. The batch file should be run from the same directory as the maxent.jar file. This will call up the maxent.jar file and begin the modelling runs.

Appendix S3: AUC and omission rate calculations.

We obtained evaluation AUC from MAXENT in two different ways. Whenever possible, we extracted AUC values from MAXENT output files. However, MAXENT does not automatically provide AUC values for models transferred to another region or time period. Therefore, for the masked geographically structured approach, we calculated AUC using the AUC tool available in MAXENT 3.2.17 via command-line arguments combined into batch files.

The relevant series of commands for the AUC tool is:

java -cp <path-to-maxent.jar> density.AUC evaluationlocs.csv prediction.asc [or .grd, .bil, etc.]

Given these commands, MAXENT will calculate AUC from a set of evaluation localities and a prediction file from the MAXENT output. The evaluation data must be saved as a .csv file, but MAXENT will accept several types of grids. Below is an example of a series of commands used in this study:

java -cp c:\GIS\Maxent\3.2.17\maxent.jar density.AUC
C:\GIS\Hetgrant\GeoPart\Locs\Eval_A.csv
C:\GIS\Hetgrant\GeoPart\Results\A\0.25\AGeo.asc >>
C:\GIS\Hetgrant\GeoPart\Analyses\AGeo_AUC.txt

Here we have used the command '>>' to tell the program to export the results and append them to the text document 'AGeo_AUC.txt'. If this command is left out, the AUC score will appear in the command shell below the command line.

Similar to the situation for AUCs, we obtained omission rates from MAXENT output files whenever possible but calculated them ourselves for the masked geographically structured approach. Calculating omission rate requires a few more steps. First, the user must extract the value of the prediction at each evaluation locality by using the GetVal tool in MAXENT 3.2.17. Those values are then compared to a cutoff value which represents some thresholding rule of the user's choice and are determined to be either presences or absences.

The commands for the Getval tool are:

java -cp <path-to-maxent.jar> density.Getval evaluationlocs.csv prediction.asc [or .grd, .bil, etc.]

MAXENT then will provide a list of localities and their corresponding prediction values. Below is a series of commands from our analyses:

java -cp c:\GIS\Maxent\3.2.17\maxent.jar density.Getval
C:\GIS\Hetgrant\GeoPart \Locs\Eval_A.csv
C:\GIS\Hetgrant\GeoPart\Results\A\0.25\AGeo.asc >>
C:\GIS\Hetgrant\GeoPart\Analyses\AGeo_Getval.txt

We then compiled the outputs into a spreadsheet and extracted the relevant threshold value from the MAXENT .html files manually. Using an 'If/Then' statement, we converted the values into absences (0) or presences (1) and calculated the omission rate.