**Extent of Occurrence and Area of Occupancy bulk calculator for NatureServe element ranking**

Created by Clark Hollenberg at the Colorado Natural Heritage Program, October 2024

Introduction

This script imports a .csv file with rows containing multi-species point occurrence data and WGS84 latitude, longitude coordinates. The output is a xlsx file containing extent of occurrence (EOO), area of occupancy with count of 2 x 2km grid cells (AOO), number of hypothetical Element Occurrences (EOs), and calculated rarity rank. This enables multi-species calculations which are not currently possible on GeoCAT. The projection coordinate system matches the .prj file from the EOO toolbox provided by IUCN. Calculated values match those of the toolbox, but may differ from GeoCAT by 1% at larger scales.

Description of files in folder:

* rarity\_calculator.ipynb = the script to calculate ranks
* sample\_occurrence\_points\_to\_rank.csv = provided example input dataset of point occurrence data
* RankingMetricRules.xlsx = spreadsheet with point rules for rank factors
* Cylindrical Equal Area (world).prj = spatial projection from EOO IUCN calculator

Configuring and running the calculator

This script requires a user license for ArcGIS Pro. You should be logged into your account on your computer before running the script. It also requires a python interpreter which can handle Jupyter notebooks (.ipynb), although the script could be converted to a .py file format as well. I use and recommend Visual Studio Code, which is helpful for debugging and writing code. It should be possible to run the script directly in ArcGIS Pro, but I haven’t tested that yet. I think it will be faster to run externally than in ArcGIS Pro. If using Visual Studio Code or some other code editor to run the script, you need to use a python interpreter that has the arcpy and pandas libraries. The easiest option is to find the python.exe interpreter that was installed with your ArcGIS Pro. On my computer it’s in C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\python.exe You can also clone this environment into a location that is easier to find.

After your python environment is setup, all you need to do is enter the file paths to your input csv and the output file paths where you want to store the calculated ranks. You can also adjust the separation distance for EOs if desired. Note that it can take several hours to run the script if you are working with very large datasets with many unique species. It takes a while to import the arcpy library for the first time, but after everything is initialized it usually takes about 5-10 seconds per species. You may want to test a smaller dataset to make sure everything is working properly before passing a large point dataset to the calculator.

Ranking Rules

Since this bulk calculator is designed to work without manual biologist review, it only considers rank factors in the rarity category. I used Range Extent (EOO), Area of Occupancy (AOO), and Number of EOs to assign a preliminary rarity rank to each species. Note that the RankingMetricRules.xlsx sheet assigns points from the corresponding “…Score” column if the calculated value for the particular metric is below the number for that row in the “…Val” column. The RankVal and RankScore columns use the same process to assign a rarity rank based on the points assigned by the rank factors.

For the ranking rules, I used the RULES sheet within the Element Rank Estimator excel macro workbook from NatureServe at https://www.natureserve.org/products/conservation-rank-calculator/download. Note that AOO is double weighted. There appears to be an error in the Step2/3 AOO values. Based on the Calculator Form, A should be assigned to 1 4km cell and B should be assigned to 2 4km cells, and C should be assigned to 3-5 km cells. I used these bin divisions and applied the points listed on the RULES sheet.

Infraspecific ranking

When ranking a binomial species without infraspecific epithet, I clustered any infraspecific taxa under the species for ranking purposes. Ex: The rank for *Abies lasiocarpa* would include occurrence points from *Abies lasiocarpa* and *Abies lasiocarpa var. bifolia*, whereas trinomial names like *Abies lasiocarpa var. bifolia* would be ranked only considering exact matches (excluding occurrences identified as only as Abies lasiocarpa).

Input data for ranking

The input csv should include at least three columns:

* SNAME – species name
* decimalLatitude – decimal degrees in WGS84
* decimalLongitude – decimal degrees in WGS84

You can change the names of these columns, but if you do they must also be modified in the python script.

I downloaded statewide occurrence data from SEINet and iNaturalist research grade. Note that occurrence data downloaded directly from GBIF may have locations obscured for certain species. A SEINet login with special permissions allows us to use the most accurate locations available. I used many different tools to clean the source data and translate the scientific names to SNAMEs used in Biotics. I won’t cover the specifics of that process here.

I provided a sample\_occurrence\_points\_to\_rank.csv file which is a slice of the iNaturalist and SEINet data that I prepared for Colorado. This can give you a sense of the fields I used and the formatting.

Estimating Number of EOs

I used a 1km separation distance for clustering plants into hypothetical EOs. For invertebrates, we used a 5km distance. This can be adjusted at the start of the script based on the taxonomic group in question.