**Information Sheet: QGIS Models & R Scripts**

**Maddie Tango**

**3/18/21**

**Necessary Materials:**

* QGIS NALCMS preprocessing model (“preprocessinglandcover.model3”)
* QGIS model (“thesismodel.model3”)
* R script: landscape metrics (“Thesis\_R2.r”)
* R script: random forests and GLM (“rf\_example.r”)
  + Made-up data spreadsheet (“RandomForests\_randomData2.csv”)
* Final Calculations spreadsheet (“FinalCalculations5.xlsx”)
* Necessary layers

**Overview**

Using the methods in Figure 1 (with some exceptions explained below) from Peters et al. (2020), I created a model in QGIS (Figure 2) and a script in R that can be applied to any wind project, provided the user uploads the correct layers (in a reasonable projection) for the particular site they want to analyze. Given land cover, wind turbine, rivers/streams, roads, and elevation inputs, the QGIS model and R script give outputs measuring landscape metrics. These landscape metrics can then be analyzed to determine how significant they are in affecting bat fatalities, using a Random Forests analysis, a generalized linear model, and Akaike’s information criterion adjusted for small sample size (AICc).

Except for the minimum distance metrics (e.g. how far the turbines are from the closest river/stream), which were calculated once, the analysis operates at four spatial scales: local, 2.5 km, 5 km, and 25 km. This is necessary because we do not know for sure what spatial scale (or combination of spatial scales) is important to bats when they decide where to roost, feed, or migrate.

At each spatial scale, the QGIS model calculates the point density of the turbines in the landscape, and the linear densities of roads and rivers/streams in the landscape.

The QGIS model also creates three layers at each spatial scale that then become inputs to the R script.

1. A land cover raster is aggregated by class type (for example, all of the forest classes become one class)
2. A forest cover raster is derived from the reaggregated land cover raster
3. A topographical position index (TPI) layer is derived from a digital elevation model (DEM)

When these layers are input into the R script, a variety of landscape metrics are calculated that measure the makeup of land cover, including clumpiness, evenness, average patch size, and percent area of each land cover class. The R script also calculates the core area of the forest class and the percent area of ridges and valleys. These landscape metrics help to understand the layout of a given landscape with the hope that particular metrics may signify risk levels for bats.

In order to use these metrics to calculate bat fatality risk levels, you would have to run the model on a decent sample size of sites. You’d then have to run your own regression using the outputs, or if you used sites in the Northeast US, you could use the coefficients calculated by Peters et al. (2020) in their regression to calculate a risk value. This risk value is dependent on surrounding sites, however, as some variables require a transformation that uses calculations from other sites. Moreover, Peters et al. (2020) did not publish which specific sites they used, so using their regression coefficients may not be accurate for the sample you are using. **BE MORE SPECIFIC HERE AND INCLUDE AS A PART OF METHODS (WRITE TRANSFORMATION)**

**Differences in Methods**

There were a few differences between the methods I used and the methods used by Peters et al. (2020).

Peters et al. (2020) used QGIS to derive the TPI and ArcGIS for all of the other GIS calculations, whereas I used QGIS for all GIS calculations. Moreover, instead of using Fragstats, I used the R package equivalent (‘landscapemetrics’) in RStudio.

The QGIS model currently operates using 6 land cover classes because the Beech Ridge (WV-2) site I used to develop this model does not have any lakes within 25 km. When I apply the model to other sites, I will replace the water class with two lake classes, large (>500 m2) and small (<500 m2).

I was also unable to find an artificial path layer, so I omitted the step where I’d need to merge artificial paths and rivers/streams into a new layer.

Peters et al. (2020) report that they used spatial joins to calculate minimum distance metrics. Instead of using spatial joins, I created proximity rasters, where each pixel stored its distance from rivers/streams, and distances from rasterized turbines to calculate minimum distances from land class types. I used the local buffer around the turbines to calculate the minimum distance to the turbine of a nearby facility.

**User Instructions**

Import the following layers & transform them into a regional projection (eg. State Plane; this can be done using ‘export as’ for vector layers and the warp tool for raster layers [use Nearest Neighbor for raster with classes and Bilinear for continuous data]). You may need to download for multiple counties and use the merge tool to stitch layers together. Included below are the sources I used for the WV Turbine sample model.

* Turbine points [.shp]
  + Only the points at the site you want
  + Model will make buffers from these
  + Source: USGS
* Non-site turbine points (must at least include the closest turbine) [.shp]
  + Select points at your site then go to attribute table and click “invert selection”, export as new layer and use these points
  + With a local projection you may need to only select the ones that are closest because otherwise the transformation won’t work when you’re doing the distance calculation
  + Source: USGS
* Digital elevation model (DEM) [.tif]: may need to merge multiple to cover site area; should be imported as projection and export and save with null = 0)
  + 3D Elevation Program (3DEP) = DEM
  + Download 1/3 arc-second resolution
  + Model derives topographical position index (TPI) from DEM at 2,000 m resolution
  + Source: ScienceBase
* Flowlines (rivers/streams) [NHDFlowline.shp]
  + Source: National Hydrography Dataset
* Roads [.shp]
  + Source: TIGER
* Land cover (with lakes added) [.tif]
  + Download land cover layer (source: NALCMS), lake vector layer, and use turbine points layer
  + Run the preprocessing model (Figure 2)
    - It will reaggregate land cover types (see Appendix 1 for details), clipping at a 25 km buffer.
  + If there are lakes within a 25 km buffer of your turbines, do the following steps manually (I kept getting errors on these when including them in the model, but if you do it manually it should work):
    - Rasterize lakes (I was unable to get lakes to be 1 and no lakes to be 0 when I first rasterized, so I had lakes be 2 and no lakes be 0, and then used the raster calculator to change the 2s to 1s.)
      * + Rasterize lakes:

Fixed value to burn: 2

Assign a specified nodata value to output bands: 1

Pre-initialize the output image with value (in advanced parameters): 0

* + - * + Then run raster calculator on rasterized lakes layer to change the 2s to 1s (“Lakes01”):

("RasterizedLakes@1" = 2)\*1 +

("RasterizedLakes@1" = 0)\*0

* + - Run raster calculator using Land Cover Aggregated and Lakes01 layer to add lakes to aggregated land cover layer with value of 23
      * ("Lakes01@1")\*23 +

(1-"Lakes01@1")\*"Land Cover Raster Aggregated@1"

* + \*How to prevent lakes from being cut off??? Could adjust buffer size but with each buffer you’re probably going to have it be cut off and to just extend the size would mess up the rest of the calculations…

After importing and preprocessing layers, follow these steps:

* Run Q model (Figure 3), using the following buffer inputs (rerun for each spatial scale) [this math is because the model starts with a 2.5 km buffer and adds/subtracts from there]
  + Local: 0-2500 = -2500
  + 2.5 km: 2500-2500 = 0
  + 5 km: 5000-2500 = 2500
  + 25 km: 25000-2500 = 22500
* Use Excel/Google Sheets (example of this in “Final Calculations” spreadsheet) to divide the following 3 measures by the total area of the landscape to calculate point and line densities; each of these should be calculated at each spatial scale
  + 1. Number of turbine points (“Statistics\_turbinecount”-> Attribute table heading: “count”)
  + 2. Length of roads (“Statistics\_roadslinelength”-> Attribute table heading: “m2”)
  + 3. Length of flowlines (streams/rivers) (“Statistics\_flowlineslinelength”-> Attribute table heading: “m2”)
  + Divide each of these three metrics by landscape area (to calculate landscape area: go to the attribute table of the buffer vector & create Area field: $area
* Run R scripts
  + 1. Input the 3 layers from QGIS described above (‘inputs to the R script’) to calculate landscape metrics
    1. Aggregated land cover raster (with lakes if there are lakes in your region)
    2. Forest cover raster
    3. Topographical position index (TPI) layer
  + 2. Run a random forests and generalized linear model (GLM requires a transformation) using independent variable outputs from the model and first R script (other independent variables to incorporate are: year, turbine height, and number of turbines at site), together with the dependent variables (number of fatalities at a site) for the sites in your region of interest (this means you will need to run the model and first R script for each site)
    1. I made a sample with made-up data, given bat fatality data was not available for all sites (see Fatalities Data Spreadsheet)

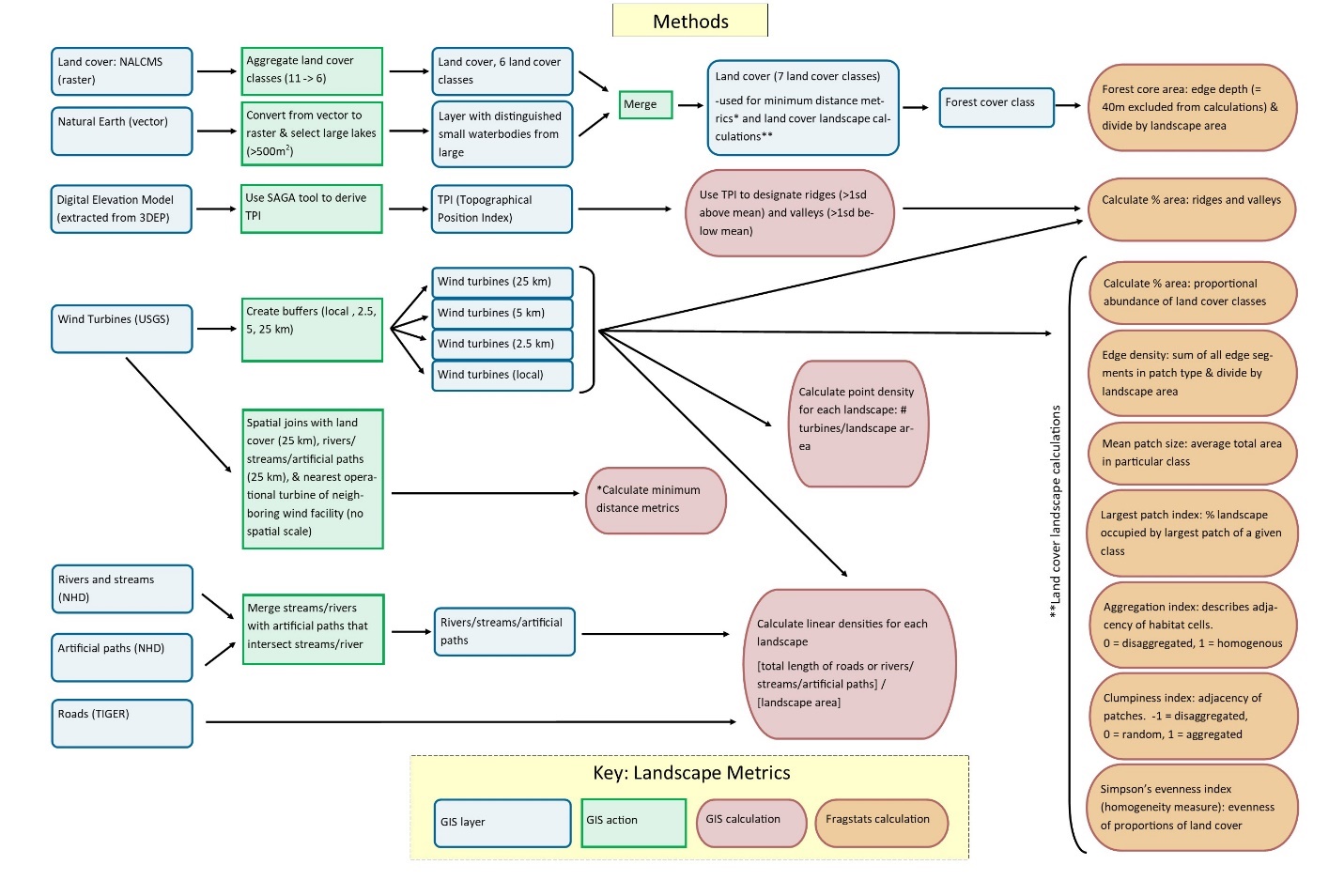


Figure 1. Workflow of independent variable calculations part of Peters et al. methods, visualized by M. Tango.

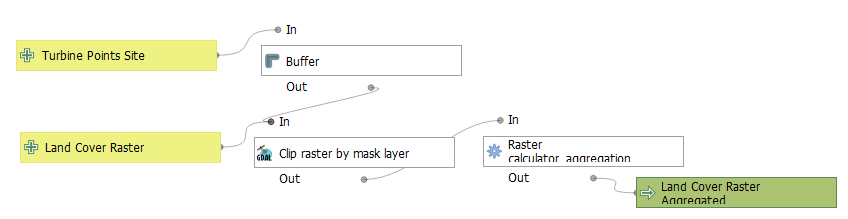


Figure 2. Preprocessing model. QGIS model: graphical modeler visualization. QGIS 3.16.4.

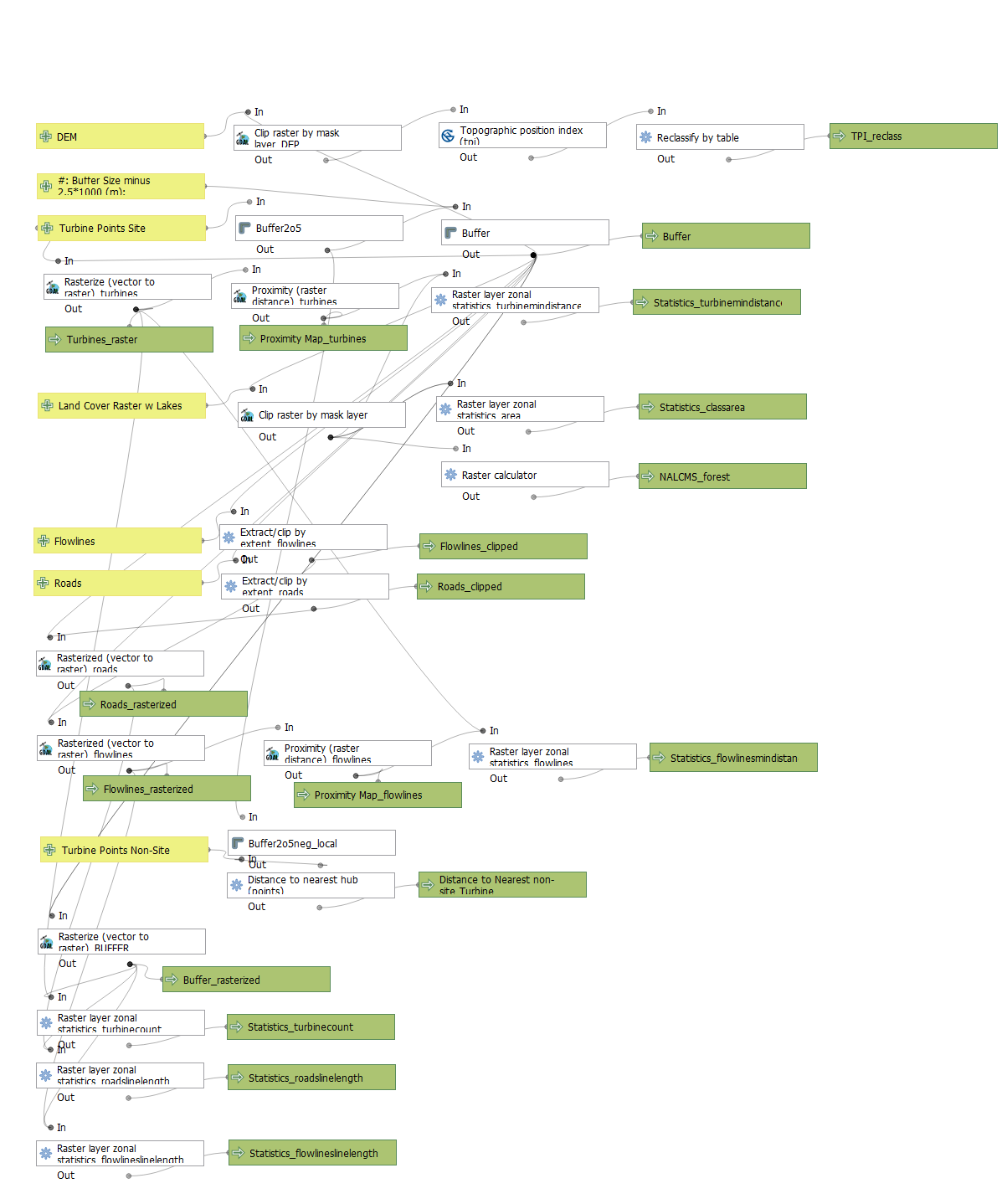


Figure 3. QGIS model: graphical modeler visualization. QGIS 3.16.4.

**Appendix 1**

NALCMS 2015 land classes are reaggregated

Layer source: cec.org/north-american-environmental-atlas/land-cover-30m-2015-landsat-and-rapideye/

* Value 14, Wetland, RGB 107 163 138;
  + Unchanged
* Value 15, Cropland, RGB 230 174 102;
  + Unchanged
* Value 17, Urban, RGB 220 33 38;
  + Unchanged
* Value 18, Water, RGB 76 112 163;
  + Unchanged (though lakes will be mosaicked in)
* Value 19, Snow and Ice, RGB 255 250 255.
  + Unchanged
* Value 20: Forest
  + Value 1, Temperate or sub-polar needleleaf forest, RGB 0 61 0;
  + Value 2, Sub-polar taiga needleleaf forest, RGB 148 156 112;
  + Value 3, Tropical or sub-tropical broadleaf evergreen forest, RGB 0 99 0;
  + Value 4, Tropical or sub-tropical broadleaf deciduous forest, RGB 30 171 5;
  + Value 5, Temperate or sub-polar broadleaf deciduous forest, RGB 20 140 61;
  + Value 6, Mixed forest, RGB 92 117 43;
* Value 21: Open Cover
  + Value 7, Tropical or sub-tropical shrubland, RGB 179 158 43;
  + Value 8, Temperate or sub-polar shrubland, RGB 179 138 51;
  + Value 9, Tropical or sub-tropical grassland, RGB 232 220 94;
  + Value 10, Temperate or sub-polar grassland, RGB 225 207 138;
  + Value 11, Sub-polar or polar shrubland-lichen-moss, RGB 156 117 84;
  + Value 12, Sub-polar or polar grassland-lichen-moss, RGB 186 212 143;
  + Value 13, Sub-polar or polar barren-lichen-moss, RGB 64 138 112;
  + Value 16, Barren lands, RGB 168 171 174;
* Value 23: Lakes
  + You add these yourself (see above)