**1:24K Hydrography Creation Toolset, in development**

**Science Services, Fisheries & Aquatic Research**

**February 26, 2013**

**Aaron Ruesch, Diane Menuz , and Matt Diebel**

**Background**

We created a hydrography database of channel, riparian area, and watershed attributes to aid in Fisheries and Aquatic Research modeling efforts. To help maintain that database with future changes in the Wisconsin hydrographic databases, we also created an ArcGIS toolbox. This document is a tutorial for the ArcGIS toolbox formatted as a walkthrough using a sample dataset.

The provided sample data is split into three folders. **/sampleData/inputs** has data that will allow you to work through the tools sequentially from start to finish. **/sampleData /outputs** has the important output data from each step of the process. If you are unable or uninterested in running some of the initial steps, you can pick up in the middle of the tutorial by using the appropriate files in this folder. You can also use these files to check the validity of your output when working through the sample data. Substitute your own output folder name in tutorial instructions so you do not overwrite existing files in the **outputs** folder. The last folder, /**sampleData /appendixData**,has files that can be used with the tools listed in the Appendix, which are not part of the main process outlined in this tutorial.

**Requirements**

1. Windows Operating System (preferably on a 64-bit machine but not required)
2. ArcGIS Desktop with ArcInfo license (for Feature Selection tool only) and Spatial Analyst Extension
3. Ability to install other freely available software when necessary as indicated in documentation for certain scripts below
4. moderate-to-high level of knowledge of ArcGIS and Python
5. moderate-to-high level of knowledge of common hydrography datasets and data structure

**Preparing Hydrography**

*Checking Network Topology*

The basic ingredients for creating a hydrography database are flowlines and a digital elevation model (DEM). Open ArcMap and add these two datasets (**/inputs/dem.img**, **/inputs/flowlines.shp**). We will be creating hydrography within a set of 12-digit HUCs on the western shore of Lake Michigan. Add the HUC12 layer to your ArcMap document (**/inputs/huc12.shp**).

The first thing we need to do is ensure valid network topology of the flowlines. In other words, all segments in the network must be connected precisely on their end nodes, and their directionality must always point downstream toward the outlet. Add the **FLoWS[[1]](#footnote-1)** toolbox to your ArcToolbox (**/FLoWS\_toolset/** **~FLoWs v9.3.tbx**). We will want to check topology using the **Check Network Topology** tool in **~FLoWs v9.3.tbx🡺4. Analysis**, but first we must create a network database, which the FLoWS framework terms a “Landscape Network.” Open the **Polyline to Landscape Network** tool in **~FLoWs v9.3.tbx🡺2. Create Landscape Network**. Under **Polyline Shapefile**, add **/inputs/flowlines.shp**, and then define your output as **/outputs/flowlines.mdb**. You must include the .mdb in the landscape network name. Once the tool has completed, open the **Check Network Topology[[2]](#footnote-2)** tool and add the **outputs/flowlines.mdb/edges** and **outputs/flowlines.mdb/nodes** feature classes from the landscape network you just created under the appropriate fields, leaving other inputs as default. Once the tool has completed, add the **nodes** feature class to your map and create categorical symbology on the **node\_cat** field with only the values **Outlet, Converging stream,** and **Diverging stream**.

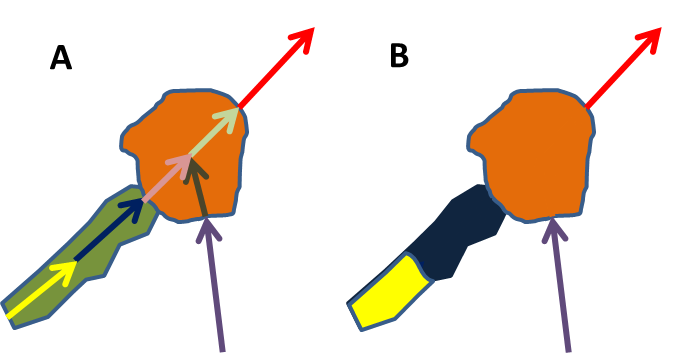
To diagnose topological problems, first check to make sure any nodes classified as **outlets** are truly outlets—if not, topology is erroneous. Next, check to see if you have any **converging** or **diverging streams[[3]](#footnote-3)**. Diverging streams may not be erroneous, but they are not supported in these tools and thus must be purged. The sample flowline dataset has one **converging stream**. The topological error arises because the flowline downstream of the node is pointed upstream. To first the error, first create a copy of **flowlines.shp** called **/outputs/flowlines\_fixed.shp**. Start editing **flowlines\_fixed.shp**, navigate to the incorrect reach, double-click it, then right-click it, and select **flip**. Rerun the Create Landscape Network and Check Network Topology tools after fixing any errors in the flowlines because you will need a topological valid landscape network in later tools. Save this final network as **/outputs/flowlines.mdb**, deleting the old landscape network file.

Other topological problems may arise, but topological editing is outside the scope of this document. The best way to avoid topological problems is to use a dataset that has been thoroughly quality checked, such as NHD, NHDPlus, or other official hydrography such as those sometimes housed at state agencies. See the Appendix for information on additional tools that may be used to prepare your hydrography and for issues that may need to be addressed in order for the tools in this document to operate as intended.

**Creating Riparian Buffers and Delineating Watersheds**

*Feature Selection*

The next step of the process is selecting which flowlines and waterbodies you want to use to create “seeds” and appropriately attributing flowlines and waterbodies with unique identifiers. If a lake is going to be a seed, then the unique identifier of the lake must be transferred to the underlying flowlines. “Seeds” are the raster representations of streams and lakes. They are called “seeds” because the watershed delineation algorithm “grows” outward from the seeds. In many cases, there will be a one-to-one relationship between flowlines and seeds; however, several flowlines that are associated with a single lake may have a one-seed to many-flowline relationship. Because this is an important concept, we have included a figure to help explain the relationship between seeds, flowlines and waterbodies (Figure 1).



**Figure 1.** Waterbody (polygon) and flowline (arrow) features in an example dataset (A), with each color representing a unique feature identifier. Raster data similar to B would be output by the **Seed generation** tool if the orange waterbody was considered a seed and the green was not. Unique identifiers of the green waterbody and of the flowlines under the orange waterbody would not be maintained in the output seed raster and thus these features will not have their own riparian buffers, watersheds, and attributes.

The **~HydroAttribution🡺1 Zone creation🡺 1 Feature Selection** tool requires a very specific data structure, but it is relatively easy to use the tool for more general situations. The first input for the tool is a **Temporary Work Folder**, where temporary process files will be stored. This can be any folder that you wish and the output stored in this folder can be deleted after the tool finishes running. The next input, **Input Streams File**, is a shapefile of topologically correct flowlines, which in this case will be **/outputs/flowlines\_fixed.shp**. This file needs three columns, a column of unique identifiers (which can be the FID) and two selection fields. The **Unique Stream ID Field** for our example is the **HYDROID**, but could be any user-specified unique identifier. Without getting into the details of the selection fields, here are some suggestions of how to set up the selection fields:

1. If you want all stream features included as potential seeds, create two fields that contain the value 1 for every feature. These fields will be **Stream Selection Field I and II** in the tool.
2. If you only want to attribute certain stream features, create one field that has 0s for excluded features and 1s for included features. This field will be **Stream Selection Field I**. Create a second field of all 0s; this will be **Stream Selection Field II**. Only features with a 1 in the first field will be included as seeds.
3. If you want to include all non-landlocked features and only those landlocked features that intersect waterbodies that you designate as lakes, create a field of 1s for all non-landlocked features and 0s for landlocked features; this will be **Stream Selection Field I**. Create a second field of all 1s; this will be **Stream Selection Field II**. All features with a 1 in the first field plus all features with 0s that intersect lakes will be included as seeds.

The file **\outputs\flowlines\_fixed.shp** works with the default values in the tool using scenario 3 above; one feature has a value of 0 in the field **Select1** because it is landlocked. This landlocked feature does not intersect a waterbody, so it will not become a seed.

The **Input Waterbody File** is a shapefile with areal water features, such as lakes, double-line rivers, and reservoirs; in our example, this will be **/inputs/waterbodies.shp.** The waterbodies shapefile must have a field of unique identifiers that are distinct values from the unique identifiers in **/output/flowlines\_fixed.shp**. The second field has values of 1 for features that will be lake seeds (see discussion above and Figure 1) and otherwise a 0. You may want to determine what qualifies as a lake in your study. based on feature type and/or feature area. The default values for the **Unique Waterbody ID Field** and **Waterbody Lake Field** are already set up to work with **/inputs/waterbodies.shp.**

The next inputs into the tool are names for three fields that will be added to the output. These fields cannot already exist in the input shapefiles. The **Network Column name** is the name that will be given to the field that indicates whether or not a feature is landlocked (e.g., an isolated non-contributing feature versus a river)[[4]](#footnote-4). The **Seed Name** is the name that will be given to the field of unique seed identifiers. The **Aggregate Name** is the name that will be given to the field that will contain the identifiers of features that can be aggregated together into confluence-bounded segments. We will use the default values in our example. **REACHID** will be an important identifier of features throughout this tutorial.

The **Study Area** input is a polygon shapefile of the boundary of the study area. In our example, this is **/inputs/HUC12.shp**, the HUC12 catchment boundaries surrounding our study area. The next five tool inputs are output locations for the outputs from the script. These can be any user-specified shapefiles. In this example, these files will be **/outputs/lakes.shp**, **/outputs/networkWB.shp**, **/outputs/excludedWB.shp**, **/outputs/finalFlowlines.shp**, and **/outputs/vertex.shp**.

The **Input dissolve feature** is a polygon shapefile which is intended to be the boundary of the Great Lakes HUC12s, but could potentially be any HUC12 boundary where the HUC12 is entirely filled by a lake (rather than including both a lake and the surrounding immediate catchment area). These features are the only features where watershed boundary delineations are allowed to cross HUC12 boundaries. The tool will identify all flowlines and waterbodies under the dissolve feature and edit out most of them[[5]](#footnote-5), so the dissolve feature should be at the end of a stream network and not in the middle. We will use**/inputs/LakeMichigan.shp** for the input dissolve feature*,* which is the HUC12 boundary for Lake Michigan. The **dissolve column name** is the name of the field that will be added to the flowlines shapefile to indicate whether a feature is under the dissolve feature. This field name cannot exist elsewhere in the flowlines shapefile.

Using our sample data, we now have two lake features that will be used as seeds and eight non-lake, non-landlocked waterbodies that will not be seeds but will be used to inform the size of stream seeds. We have lost several flowlines, including the landlocked feature and features entirely under **/inputs/LakeMichigan.shp.** Both the lake and the flowline files have a **REACHID** field and the REACHIDof the flowline under the lake matches that lake’s REACHID.

*Create a shapefile of coastlines* *for very large waterbodies*

A coastline layer is necessary to allow overland flow to large waterbodies such as Lake Michigan. This layer will help 1) delineate a single multipart watershed for waterbodies that span multiple HUC12s and 2) aid in accurate delineation of the tributary/waterbody interface for very large waterbodies. The best way to create this feature is to takeconvert **/inputs/LakeMichigan.shp** (or any feature used as the **Input Dissolve Feature)** to a polyline using **Data Management🡺Features🡺 Polygon To Line**. You can then edit out any features outside the scope of your analysis. For the sample data, this layer has already been created for you as **/inputs/greatLakesCoast.shp.**

*Seed Generation*

This tool takes the output from the **Feature Selection** tool and creates a rasterized seed. The seed raster is composed of flowlines and lakes, with flowlines expanded to fill in the area of any non-lake waterbodies that are underneath them. Set the **Temporary Work Folder** to a folder location where temporary work can be stored. Set the next four inputs to **/outputs/vertex.shp**, **/outputs/networkWB.shp**, **/outputs/finalFlowlines.shp**, and **/outputs/lakes.shp**, four of the outputs of the **Feature Selection** tool. The **Unique Seed ID Field** is a field shared by the input streams and lakes that uniquely identifies seeds; in our example this will be the field **REACHID.**

The **Input Coastlines** (as described in the previous step) should be a polyline version of the feature(s) used as the dissolve feature in the **Feature Selection** tool. This coastline feature serves as a seed for features like Lake Michigan which can receive direct overland flow across HUC12 boundaries. The shapefile must have a field called **HYDROID** populated with the values **600000001** or **600124406**; this is currently hard-coded into this and subsequent tools. If you do not want to include a coastline feature in the analysis, then create a polyline shapefile that does not intersect the study area. This will allow the script to run correctly without influencing the actual output. Add **\inputs\greatLakesCoast.shp** to the tool.

The **Input Elevation Grid** should be a DEM file that covers at least the area of interest; if it is larger, it will quickly be clipped to the study area. We can use **/inputs/dem.img** with the sample data. The **Input Boundary** should be a polygon shapefile that covers the extent of the area of interest; in this example we use the HUC12 boundaries, **/inputs/HUC12.shp**. Set the outputs to **/outputs/seeds.img** and **/outputs/vertex.img**.

This tool may not run and/or may take a prohibitively long time to run if the study area is too large. If this is the case, then it would be advisable to break the study area into separate processing units and to then run each unit separately. You do not have to break the input files (e.g. streams, lakes, etc.) into separate processing unit files as long as you use a **Input Boundary**; the tool will select out only those features that are within the boundary for the rest of the analysis. For the Wisconsin project, the largest processing unit that we used was 10876 km², though the exact limitation of the tool will depend on a combination of boundary size, data resolution, and number of features present within the boundary.

*Delineate Watersheds*

Once seeds and vertices have been created using the **Seed Generation** tool, we are ready to delineate watersheds. Open the **Delineate watersheds** tool in **~HydroAttribution🡺1 Zone creation**. Under catchment features, add **/inputs/huc12.shp** using the **HUC\_12** field as the **Catchment Redelineation Unit Field[[6]](#footnote-6)**. These two arguments will define processing units for delineating watersheds—that is, the tool will loop through each unique value in the **HUC\_12** field and run an independent delineation routine on each. Add the outputs of the **Seed Generation** tool, **/outputs/seeds.img** and **/outputs/vertex.img** to the **Input Seed** and **Input Vertex[[7]](#footnote-7)** arguments. Add **/inputs/dem.img** to the DEM argument. This is a raw DEM, so if you would like the DEM to be conditioned (i.e., sinks filled, streams burnt, walls built[[8]](#footnote-8)), check the **Do DEM Conditioning?** checkbox. Checking this box is strongly advised, as there will be many inconsistencies in the output without it. Last, set **Output Watershed** to **/outputs/watersheds.shp.**

The output watersheds will conform to the input HUC\_12 boundaries and have values in the field **CATCHID** that corresponds to the unique identifiers used in the **Feature Selection** script (in our example, to the **REACHID** field). Users may wish to dissolve the output based on the **CATCHID** because features may occasionally cross the HUC12 boundaries when input features are not consistent with HUC12 features. There are fewer features in **/outputs/watersheds.shp** than there were unique **REACHID** values; this is because some features are too small to become seeds and/or to have watersheds delineated.On occasion, the delineations of the output watersheds will not completely cover the study area. In other words, small gaps between or within watersheds may exist. This is due to a relatively complex problem that is outside the scope of this document. When we used these tools to create watersheds for Wisconsin, our solution to this problem was to use “Cost Allocation” (an ArcGIS tool in the Spatial Analyst extension) to fill in the gaps. We included a beta version of this tool in **~HydroAttribution🡺Miscellaneous tools**. Check to make sure the output watersheds have a defined projection; if not, use **Data Management Tools🡺Projections and Transformations 🡺Define Projection** to set the projection as the same projection as the DEM used for conditioning.

*Create Riparian Buffers*

This tool is almost as simple as it sounds. We need to delineate areas that are within a specified distance from each of our water features. It is nearly as simple as creating polygon buffers around features, however this tool deals with allocation of area near confluences in a special way. Add **/outputs/seeds.img[[9]](#footnote-9)** and  **/outputs/vertex.img** to the **Seed raster** and **Vertex raster** arguments. Then define a **buffer size** (in the same units as your coordinate system) and an **output raster**. We will save the output raster as **/outputs/riparianBuffer.**img. Riparian buffers will have the same cell size as the input data and therefore the realized buffer size may differ from the input buffer size if the input buffer size is not evenly divisible by the cell size of the input data. In other words, if the input is 10-m resolution and the selected buffer size is 25-m, actual buffers will be either 20 or 30-m in width. This tool may take a long time to run depending on the scope (i.e., extent and resolution) of your study area[[10]](#footnote-10).

*Build Watershed/Riparian Topology*

We now have watersheds created; however they are simply polygons that have no knowledge of their relation in flow, which we will refer to as “topology.” As a term, “topology” can have several different meanings, but here we use it describe information that relates spatial units (in our case, polygons and polylines) to their downstream neighbors.

Our Landscape network already has polyline-to-polyline topology built in (see **/outputs/flowlines.mdb/relationships**). We will conflate the polyline-to-polyline topology of the landscape network to polygon-to-polygon topology of our newly created watersheds. You must use a version of the landscape network built with clean topology in order for this to work. You first have to modify one of the files in the landscape network. Add the **edges** file to your ArcMap document; we can use **outputs\flowlines.mdb\edges** because we know this landscape network has clean topology. Next add **outputs\finalFlowlines.shp**. This is the feature that was output by the **Feature Selection** tool that has unique identifiers for each seed (**REACHID** in our example). Add a field of type Long Integer to the **edges** file; call this field **REACHID.** Next, join the **edges** file to **finalFlowlines.shp** based on the field that was the uniquely identifiers each flowline (same as **Unique Stream ID Field** in **Feature Selection** tool). This will be **HYDROID** in our example. Update the **REACHID** field in **edges** with the values in the **REACHID** field **finalFlowlines.shp**. You have now populated the edges field with unique seed IDs for each feature. Remove the join between features. Update any null values in the **edges** **REACHID** field to 0 or some other value that is not a unique seed ID; these features were pruned from the study area by the **Feature Selection** tool.

This tool requires special numeric software that must be installed prior to runtime. The first software package is called **numpy**, which is a “module” that can be plugged into Python. The Python build that comes packaged with ArcGIS already has numpy installed, however the tool requires version 1.6 or later, which can be downloaded at:

<http://sourceforge.net/projects/numpy/files/NumPy/1.7.0/numpy-1.7.0-win32-superpack-python2.6.exe/download>

Once numpy has been installed, we need to install an additional numeric module called **pandas**. An executable to install Pandas can be downloaded at:

<https://pypi.python.org/packages/2.6/p/pandas/pandas-0.10.1.win-amd64-py2.6.exe>

Now that the software packages are installed, we can run the tool. Open **~HydroAttribution🡺1 Zone Creation🡺5 Build Topology Using Pandas**. We are first going to create topology for the watersheds; add **/outputs/watersheds.shp** to the **Zone** argument. Add **/outputs/flowlines.mdb** to the **Landscape network topology** argument. The **Landscape network topology** argument provides space for multiple landscape networks in cases where flowline feature classes are too large to be lumped into a single landscape network. The **Seed ID** **field** specifies the field in the **edges** feature class that matches the IDs of the feature in the **Zone** argument. This field will have unique seed IDs; in our example, we will use **REACHID.** \*Important\* If you are creating topology for watersheds, the IDs that match the **Seed ID** field in the landscape network must be stored in a column called **CATCHID** in the watershed shapefile; this is the default output of the **Watershed Delineation** tool and so we do not have to make any changes with our example data.

If all your watersheds are on a drainage network (i.e., **not** landlocked/isolated), then the tool is ready to run with the above parameters. However, if your watersheds include watersheds that are NOT on the drainage network (i.e., landlocked/isolated features such as “seep” lakes), then topology must be created for these landlocked features using a non-conflation method independent of landscape-network topology. Otherwise, the output of landlocked features will be indicated with a placeholder value of 99999. The method used in this tool creates topology based on terrain, and thus a filled[[11]](#footnote-11) **flow direction** raster must be specified if landlocked features occur[[12]](#footnote-12). Creating a **flow direction** raster requires manual processing outside the framework of this toolbox. A sample has been provided for you in **/inputs/flowDirection.img**. Information on creating your own flow direction raster can be found by searching the term “An overview of the hydrology toolset” in the ArcGIS Desktop Help menu. We will select **Connect Isolated Feature** and use **flowDirection.img** because we have one isolated lake in our dataset. Finally, define an output topology dataset. If the **Zone** argument is a shapefile, you must add the .shp extension to the name of the output dataset. In our example, our output dataset is **/outputs/watershedsTopology.shp**

The output will be exactly the same dataset as your newly created watersheds with the addition of a new field, **TOCATCHID,** which contains the **CATCHID** of the downstream feature, or 999999 for features that do not flow to existing watersheds. See discussion of **Circular Topology Table** under *Trace Attribution****,*** below, for information on topological problems that can occur when creating topology for isolated features.

Next create topology for the riparian buffers. The Value in the riparian buffer raster needs to match the **Seed ID** **field** in the landscape network. The tool is currently not set up to connect isolated riparian buffer features to the larger network, so do not select **Connect Isolated Feature**. This makes sense for riparian areas since riparian areas that are not actually connected on a network would not be expected to strongly influence a site.

**Creating Value-added Attributes**

At this point, we should have the spatial framework for building our hydrologic database. Now we want to add value to the database by creating new attributes that represent watershed or riparian-area characteristics. We can represent these characteristics “incrementally” (of the watershed or riparian area containing the feature) or “cumulatively” (of the watersheds or riparian areas containing all upstream features). We will go through processes for each, both with continuous-type data attributes (e.g., slope, drainage area) and categorical-type data attributes (e.g., land-use, soil type). But first, some data preparation.

*Convert zones to raster*

Value-added attributes are all derived from summarizing raster grid values within raster “zones” (i.e., watersheds or riparian areas). First we must create a raster zone layer with unique site IDs as the **Value**. If you are attributing riparian areas, fortunately the riparian areas layer should already be in raster format. However, if you are attributing watersheds, we must convert the watershed polygons to a raster within the same domain (extent, resolution, and grid alignment) as the DEM in IMAGINE format (.img extension). Make sure that **/outputs/watersheds.shp** has a projection defined; if not, define the projection. Next, add a long integer field to **/outputs/watersheds.shp** called **CATCHID2.** Copy the values from the string **CATCHID** field into this new field using the field calculator. Open the **Polygon to Raster** tool in **ArcToolbox🡺Conversion Tools🡺To Raster**. Add your watershed shapefile (**/outputs/watersheds.shp)** as an **Input Feature** and use **CATCHID2** as the **Value field**. Store the output raster as **/outputs/watersheds.img**. The **Cell assignment type** is your choice—see the ArcGIS help menu for details on each method. Leave **Priority field** as **None**. Finally, choose **/inputs/dem.img** for **Cellsize**. To ensure the output grid is in the same domain, also set **Environments🡺Processing Extent🡺Snap Raster** to **dem.img**.

*Raster resampling*

Calculating zonal statistics requires a zone dataset and underlying raster dataset that are in precisely the same domain (i.e., extent, resolution, grid, coordinate system). The **Raster resampling** tool in **~HydroAttribution🡺2 Attribution** coerces raster grids (zones and underlying raster datasets) into the same resolution and grid as the DEM used for delineating watersheds, and the same extent as the zone dataset. The tool will loop through all rasters in a folder, which is an efficient way to resample rasters when there are a large number of rasters to deal with. First, define a **Raster data folder** which contains all raster datasets (not including the zonal spatial layer)—the raster datasets must be in IMAGINE format (.img extension) and only datasets of these formats can be stored in this folder. You can use your own raster datasets or those provided in **/inputs/attributeRasters**. Then, define an **Area boundary** which should be the extent of all area you want included in the output rasters. With the sample data, you can use **/outputs/watersheds.shp** . Finally, define **/inputs/dem.img** as your DEM, define an **Output folder** where resampled rasters will be stored, and click **Ok**. We will store the output rasters in **/outputs/rastersResamp**.

If you also need to reproject your raster into a new projection, you can instead use the ArcGIS tool **Data Management Tools 🡺 Projections and Transformations 🡺 Raster 🡺 Project,** but you will only be able to reproject and resample one attribute tool at a time. Make sure you set the output cell size as **/inputs/dem.img** and select Environments at the bottom of the tool in order to set the **Processing Extent 🡺 Extent** as the zonal data and the **Processing Extent 🡺 Snap Raster** as **/inputs/dem.img.**

*Cross-tabulated and zonal statistics of raster attributes*

Now we are ready to create value-added attributes for our hydrography database. The first step is to summarize raster datasets within each zone (either watersheds or riparian buffers), using one of two methods: cross-tabulated area summaries (for categorical data) and zonal statistics (for continuous data). ArcGIS has tools to run these calculations; however, the tools will not work or will take a prohibitively long time with very large study areas at fine resolution. We have built the tools **2a Crosstab using R** and **2b Zonal stats using R** tools in **~HydroAttribution🡺2 Attribution** to perform these calculations more efficiently. The names of the tools imply that we will be using R software to perform these calculations (R performs better with very large zonal operations). Therefore, you must install the software before proceeding (skip ahead if you already have it installed on your machine).

Download the latest version of R statistical software using the following link: <http://www.r-project.org/>. From here, click on the mirror site that is closest geographically. Then, click **Download for Windows**, then **base**, then **Download R 2.X.X for Windows**. Install using the downloaded executable (note the installation directory when prompted). In addition to R, the tools require special spatial packages that plug in to R, primarily **GDAL** and **PROJ.4** which are sets of libraries that define standard geographic data formats and reference systems. The easiest way to install these libraries is by installing **FWTools** (open-source geoprocessing software) using the link to the executable at <http://fwtools.maptools.org/>. Once FWTools is installed, we need to install the **rgdal** and **raster** packages for R. The easiest way to do this is by opening the R GUI and pasting the following line of code at the prompt:

install.packages(c("rgdal", "raster"))

Once the packages have installed, you can close the R GUI. You’re now ready to run the attribution tools.

Let’s start by running the **2b Zonal stats using R** tool. We have two example continuous datasets- **adjPermeability.img** and **percentImpervious.img**. Using these datasets, we can calculate the mean percent imperviousness and mean adjusted permeability of each of our watershed or riparian zones. The attribution tools work by calculating data for all rasters within a specific folder, so we need to create separate folders for our categorical and continuous attribution rasters. Create folders called **zonal** and **crosstab** under **/outputs/rasterResamp** and move **percentImpervious.img** and **adjPermeability.**img to the zonal folder and **landCover.img** to the crosstab folder. Open the **Zonal Stats Using R** tool. The **R Directory** will likely need to be changed. Find the **bin** directory within your R install directory that contains the file **Rscript.exe**. Next, define the **Zonal stats R script** by navigating to the directory where you stored this toolbox, and selecting **zonalCalledFromToolbox.r**. **Raster folder** should be **sampleData/outputs/zonal** and **Zone dataset** for this example will be the watershed raster you just created, **\outputs\watersheds.img**. Define **/outputs** as the **Output Folder** and a **Scratch Workspace**. If working with extremely large zones, large temporary files will be saved in the scratch workspace. The scratch workspace should therefore have a large capacity and files within it may need to be deleted periodically to clear up memory. Our sample area is small enough to not require any temporary files to be created. Under **Run ID**, define an optional suffix to be concatenated to output table names. Using unique Run IDs each time you run the tool will prevent files from being overwritten in your output folder. Similarly, **Zone Type** concatenates information to the output table name. You may want to use something like “W” to denote watershed attributes and “R” to denote riparian attributes. Under **Zonal Statistic**, define the statistic that will be calculated across all cells that lie within each zone. We will select the default, mean, for our sample data. Finally, under **Raster format**, choose the type of rasters, either TIFF or IMAGINE (**.tif** or **.img** file extensions) that are stored in the **Raster folder** (ensuring that the R script can find the correct raster inputs). You can also run this tool for both the channel features (**/outputs/seeds.img**) and riparian buffers (**/outputs/riparianBuffer.img**).

The inputs to **2a Crosstab Using R** are nearly identical to the zonal stats tool described above. Change the **Raster folder** to the folder of categorical raster data, **sampleData/outputs/crosstab**. The **Zonal Statistic** does not need to be defined. Instead, the tool by default calculates the number of pixels of each unique value of each input raster in the **Raster folder** and reports those numbers under a column corresponding to each unique value. If there are many unique values in many input rasters, the output table can be rather large, so it is advisable to compute the size of the output attribute table before running this tool.

The output for both of these tools is the csv file **finalData\_(*value of Run ID*).csv** with a **CATCHID** field matching the unique zone values of the input and a **cellCount** field indicating the number of pixels within the zone.[[13]](#footnote-13) **cellCount** can be converted to an area, representing the watershed or riparian area, by multiplying the cellCount by the area of each cell. Since our sample data has a cell size of 10 m, you can obtain the area of each cell by multiplying the cellCount by 100 m². For the zonal tool, each remaining field represents a single input attribute raster. The field name will be the **Zone Type** followed by the name of the attribute raster and the **Zonal Statistic** used, such as **W\_adjPermeability\_mean**. For the crosstab tool, each remaining field contains the value of a particular input raster and unique value combination, with field names formatted as (***Zone Type)*\_(*name of attribute raster*)\_(*unique value****)****.*** There is an additional field added for each attribute raster with the unique value of **NA**. This field will indicate how many, if any, cells in a zone had a missing value in the underlying attribute raster. The files **/outputs/finalData\_W\_crosstab.csv** and **/outputs/finalData\_W\_zonal.csv** are outputs from running the crosstab and zonal tools on the watersheds in our sample data.

*Trace Attribution (or ‘cumulative’ attribution)*

So far we have only calculated attributes for zones associated with each individual seed feature (AKA ‘incremental attribution’). In hydrology, we are often interested in cumulative attributes that summarize all zones that drain into each feature. This is called **Trace attribution** because for each feature, the tool traces the upstream network to find associated upstream zones.

To run this tool, make sure that **numpy** and **pandas** modules are installed as described in the documentation for the **Build Topology** tool.

Open the **Trace Attribution** tool in **~HydroAttribution🡺2 Attribution**. Under **Zones**, select **/outputs/watershedsTopology.shp**, which is the output from the **Build Topology Using Pandas** tool. **Zone subset field** is optional and we are going to skip it for now. Next, under **Attribute Table**, add **outputs/finalData\_W\_crosstab.txt.** This argument could by any text file output by the tools **Zonal stats using R** or the **Crosstab Using R**. Remember from the zonal and crosstab tools that the zonal statistics table summarized continuous data information (e.g., percent imperviousness) whereas the crosstab table summarized categorical data (e.g., land cover classes). The defaults for the next two fields, **Zone ID** Field and **Area Column**, will work with our input table. **Zone ID** relates to the **CATCHID** column in the **watershedsTopology.shp** layer and **Area Column** is the incremental area of each zone, and is used for calculating upstream area-weighted means. We are trace-attributing a crosstab table, so we will choose **sum** for the **Statistic.** With a zonal statistics table (continuous data), you would select **area-weighted average**. Finally, define an output table with **.txt** extension to store the trace attribution table. We will set the output as **/outputs/finalData\_WT\_cross.txt**. If you traced categorical data (i.e., from the crosstab tool), the output will equal the number of cells in the upstream drainage area for each cell value. Categorical data is often reported as percentage of total land area. To report as such, divide each column by the **cellCount** column, which stores the total number of pixels in the upstream drainage. The **cellCount** column can also be used to compute drainage area by multiplying by the cell area if the raster attributes were stored in a projected coordinate system.

Optionally, you can create a **Circular Topology Table**. A circular topology table will store all watersheds with invalid, or circular, topology where topology for two or more watersheds are presumed to flow into each other. This may be the case if you chose to **Build Watershed/Riparian Topology** with isolated features. When topology was assigned to isolated features, the process is not intelligent enough to catch these errors. They must be fixed manually. If you chose to output a circular topology table, check the output for any stored values (open in Notepad or a text editor of your choice). If there is nothing stored in the text file, there are no instances of circular topology. However, if values are stored here (always stored in sets of two or more) topology for the watersheds with the associated **CATCHID**s must be fixed. To do this, add the **watershedsTopology.shp** layer to ArcMap, open the attribute table, and select the **CATCHID**s from the circular topology table. Using an underlying DEM or topological feature map, manually interpret the correct flow topology by editing the value in the **TOCATCHID** column. After you have fixed all circular topologies, re-run the **Trace Attribution** tool. We are not going to search for circular topology in our example.

Sometimes users may be interested in only running upstream traces for certain features, such within a study area or only for lakes or certain kind of stream segments. This can save processing time in a large study area (tracing over 160,000 features in the state of Wisconsin takes about 7 hours). Let’s assume that a user wants to only find upstream data for lake features. Add a short integer field to **/outputs/watershedsTopology.shp** called **TRACE.** Sort descending on the **CATCHID** field and select those features where the CATCHID begins with a 6. These should be lakes in our study area. Use the Field Calculator to change these features’ **TRACE** field to the value 1. Rerun the cross tab trace, this time setting the **Zone subset field** t0 the new **TRACE** field that was just added. Now the tool will only output data from the features that had the value 1 in this column.

*Connectivity attribution- distance to the nearest lake*

To-from data from the landscape network can be used to determine the distance between each feature to features with attributes of interest, such as to lakes or watershed basins of specified sizes. The script **~HydroAttribution🡺2 Attribution🡺4 Nearest Lake** outputs a table with the distance between each feature to the nearest lake of a specified size, taking into account lack of connectivity due to dams. The nearest lake can be in any direction- upstream, directly downstream, or downstream and then up a tributary.

The inputs for this script include two csv tables that can be extracted from **/outputs/flowlines.mdb** and several user-defined columns to specify feature attributes. Three columns will need to be added to the attribute table of the **edges** feature in the **flowlines.mdb**, assuming that you are have already added the field **REACHID** as specified in *Build Watershed/Riparian Topology*, above. First, a field must be added that indicates whether there is a dam on a particular feature and if so, whether the dam is closer to the upstream or downstream end of the feature. Add a short integer field to **edges** called **DAMS** and update attributes for each feature to reflect presence and location of dams: 0 indicates no dam on a feature, 1 indicates dam on upstream side, and 2 indicates dam on downstream side. If there is no dam data available, attribute all features with 0 in this column[[14]](#footnote-14). We have included a mock dam file called **/inputs/dams**. Since there are only two dam features in the file, it is easiest to attribute the associated features manually. Update the field **DAMS** to 0 for all features except the two features that intersect dams. These features will receive values of 1 and 2 because the location of the dam in relation to their downstream segment.

Next, add a double field to **edges** called **LAKES**. This field will be used to indicate the area of lake features associated with particular flowline segments. We can obtain lake areas from **outputs/lakes.shp**. This file already has lake areas calculated (in acres) in the field **AREA**. If it was missing this field, it would be easy to calculate an area using Calculate Geometry in ArcGIS. Join **edges**  to **lakes.shp** using their common identifying field, **REACHID.** Update the **LAKES** field in **edges** by setting it equal to the **AREA** field in **lakes.shp[[15]](#footnote-15)**. Set any null values in the **LAKES** to 0 to indicate no lake associated with those features.

The last field that needs to be added to the edges file is a short integer field that indicates whether tracing should stop once it reaches a particular feature. In most cases, this field can be populated entirely with 0s. Features attributed with a 1 in this field will serve as the downstream-most feature that any other feature can trace to. This can be useful to save processing time if there are many features in a dataset that are not of interest to you. This could include flowlines indicating movement through the Great Lakes, if the Great Lakes themselves are not attributed or this could be an out-of-state river that is not properly attributed and thus not included in analysis. Add a double field called **STOP** to edges. Populate with the value 0. Use Select By Location to find **edges** that have their centroid in **inputs/LakeMichigan.shp**. Update the field **STOP** to 1 for the selected features and then clear the selection.

Once these fields are added to the **edges** file, the table can be exported as a csv file for analysis. Right click on edges and select Open Attribute Table. Left click on the icon on the top left of the table view and select Export from the drop down menu that appears. Navigate to the folder where you want to save the file, enter a file **name** that ends with .csv and change the **Save as type** to “text file.” Use the same process to export the **relationships** table from the **/outputs/flowlines.mdb** into a csv file. Make sure that both files are csv files. We have saved **edgesNearLake.csv** and **rels.csv** in **outputs.**

Now we are ready to run the **Nearest Lake** tool. First, update the **R Directory** to the location of the R install directory that contains the file **Rscript.exe.** This will most likely be under **Program Files\R** in a directory called **bin.** Next, navigate to the correct location to find the script **connectivityNearestLake\_CalledFromToolbox.r**. This should be stored with the rest of the tools in the toolbox. **Maximum search distance** should be a number, in the same units as the stream length (see below), which indicates the maximum distance that should be searched to look for the nearest lake. Features that do not connect to a lake within this distance will be attributed as not connecting to a lake, even though they may connect at a greater distance. Using a very large distance with a large dataset will slow down processing. We are going to use 10,000 with our sample data. **Lake Minimum Area** and **Lake Maximum Area** should be in the same units as was used to calculate the field **LAKES** in the **edges** file. The distance to the nearest lake greater than or equal to the **Lake Minimum Area** and less than the **Lake Maximum Area** will be calculated by the tool; other lakes not in the size range will be ignored. Set the minimum to 5 and the maximum to 20. Add the relationship table, **/outputs/rels.csv** to **Relationship Table**. Default values for **FROM ID column** and **To ID column** are set to work with landscape network relationship table and do not need to be changed. Set **Flowline Data Table** to **/outputs/edgesNearLake.csv.** The remaining arguments before the output all relate to field names in the **Flowline Data Table.** The default values will work with **edgesNearLake.csv**. The fields **rid** and **Shape\_Length** were created by the landscape network and are important for connecting features to their relationships and for attributing segment length. The last step is the creation of an output csv file; we will call the file **/outputs/nearestLake.csv.**

The output for this script will be a table with the fields **REACHID**, **NearLake**, **NearLakeDist**, and **maxSearchDist**. The **REACHID** field corresponds to the field input in the **Reach Field column** for the script. The **NearLake** field corresponds to the unique ID of the nearest lake to a given feature. The **NearLakeDist** corresponds to the distance to the nearest lake of the specified size class (between the **Lake Minimum** and **Maximum Area)**, in units of the input stream length field. Distances are calculated from the midpoint of the segment of interest to the nearest lake edge and are 0 if the segment is directly associated with a lake. If there is no lake within the **Maximum search distance** of the feature that is not blocked by a dam, then the **NearLake** and **NearLakeDist** fields are population with a value of “NA.” The field **maxSearchDist** represents the maximum distance searched in the script before either reaching a lake, reaching the **Maximum search distance**, or running out of places to search (e.g. because the feature is on a small network with no lakes or is blocked by dams).[[16]](#footnote-16) When there is more than one flowline associated with a single **REACHID** (i.e. multiple flowlines under a single lake), the other fields in the table are calculated as the mean of the values for all flowlines with the same **REACHID**. Any fields attributed with NA are excluded from the calculation of the mean. **NearLake** and **maxSearchDist** will not necessarily be meaningful for these features because flowlines under a single lake may be close to two different lakes.

Users may find these additional notes useful when running this tool. First, any feature that is not connected to any other feature will not have output data associated with it. Values for these features can be manually updated; the distance to the nearest lake for these features will be 0 if the isolated feature is a lake of the specified size and NA otherwise. Lakes with multiple flowlines can potentially be associated with a nearest lake that is only connected to part of the total lake if there is a dam dividing part of the lake. In order to obtain output for every individual flowline instead of aggregating data by **REACHID**, you can set the **Reach ID column** to the same field as the **Feature ID column.** This tool can be used in a more general manner to find the distance in any direction to the nearest feature with any user-specified attribute. The field used as the **Lake Area Field** can be set as 1 for features with the attribute of interest and 0 for all other features. The **Lake Minimum Area** field could then be set to 1 and **Lake Maximum Area** field to 2.

Specific instructions for using this script with data from the original Wisconsin 24k attribution project can be found in the Appendix.

*Connectivity attribution- distance to the nearest downstream feature*

To-from relationships between features can also be used to determine connectivity to downstream features, such as the distance to a watershed of a given size or the distance to a particular waterbody downstream (such as a Great Lake). The tool **~HydroAttribution🡺2 Attribution🡺5 Downstream Trace** is designed specifically to find the distance to the nearest watershed of a particular size but can also be used to run more general downstream traces to other user-defined features of interest.

The first step in running this script is creating two input csv files. We are going to assume that the user already created a file **/outputs/rels.csv** and made changes to **/outputs/flowlines.mdb/edges** as described in *Connectivity attribution- distance to the nearest lake*, above. We are going to add a watershed area field to **edges.** Add a double field to **edges** called **SHEDAREA.** There are several ways we could update the watershed area field; we are going to use the file **/outputs/finalData\_WT\_crosstab.txt.** This file has a field, **TrcellCount**, whose value is equal to the number of raster cells in the entire drainage area of a feature. Join the **REACHID** field in edges to the **CATCHID** field in **finalData\_WT\_crosstab.txt.** The watershed area in square kilometers is equal to **TrcellCount**/10,000 because the resolution of the watershed cells is 10 m². After making this calculation, remove the join between features. Export **edges**  as a csv file with the name **/outputs/edgesDownstream.csv**.

Now we are ready to run the script. First, update the **R Directory** to the location of the R install directory that contains the file **Rscript.exe.** This will most likely be under Program Files\R in a directory called **bin.** Next, make sure that the **Connectivity Script Location** is set to the proper path to **distToLargeShed\_calledFromToolbox.r**. This should be stored with the rest of the tools in the toolbox. Set **/outputs/edgesDownstream.csv** as the **FlowlinesTable.** The defaults for the next five values should work with fields in **edgesDownstream.csv**. The **Watershed Area Threshold** is a user-defined value in the same units as the **Watershed Area Field.** The distance between each feature and features with areas equal to or greater than this value will be calculated for each feature. We will use 25 km² as our threshold value. Set **Relationship Table** to **/outputs/rels.csv**. Again, the default **From** and **To ID Fields** will work with this input. Last, set the output file to **/outputs**/**downstream25kmShed.csv.**

The output for this tool will be a csv file with four fields. The **fromfeat** and **reachID** fields are values from the **Trace ID Field** and **Reach ID Field**, respectively. Output data is not aggregated by reachID, so the user may want to do this in order to have single values for features such as lakes with multiple flowlines. The **distance** field will have the distance to the nearest watershed above the **Watershed Area Threshold** size in the same units as the **Length Field**. Values of NA indicate that a feature terminates before reaching a watershed of the specified size. The field **dam** has the value 1 if there is a dam blocking connectivity to the nearest watershed of the size of interest and otherwise has a 0.

This script can be used in a more general manner to determine downstream connectivity to any attribute of interest. First, a column would need to be added to the edges file that contains a 1 if a feature has the attribute of interest and otherwise a 0. This field would become the **Watershed Area Field.** The **Watershed Area Threshold** would then be set to 1.

**Appendix**

Following are several components of this project that may not be relevant to many users of this tutorial. We are including documentation for several ancillary tools that have not undergone rigorous testing and which address narrow circumstances that many users may not encounter. There are three tools related to data quality checking that can be used before proceeding with the zone creation tools. These tools may help highlight potential issues with data quality that users may want to consider, even if users do not use the following tools to address those issues. One tool fixes a minor issue that may occur sometimes with watershed delineations. In addition to the tools, we have included documentation on how to run one of the connectivity scripts with the attribution database and spatial data created by this project.

*Check landlocked status*

Suppose a user wants to exclude all landlocked features from their project and has flowlines data attributed to show whether a feature is landlocked. The user may want to run a check to determine whether this attribute is generally correct. This tool was designed with that situation in mind, though there are certainly many checks that a user could run.

Add the file **/appendix/flowlines\_fix.shp** to an ArcMap document. The field **LANDLOCK** is coded with 1 for features that are landlocked; however, this “raw” data is not always correct. To run the tool, we first must split the flowlines file into two shapefiles, one with features that are coded as landlocked and the other with non-landlocked features. Select By Attributes Landlocked=1, then export the selecting features to a new file called “landlocked.shp.” Switch the selection and export the resulting selection to a new feature named “notlandlocked.shp.”

Open the tool **~HydroAttribution🡺Miscellaneous Tools🡺Check landlocked status.**  The inputs to the tool are fairly straight forward. **Landlocked Features** and **Features Not Landlocked** will be the two files created above. **Output error file** is the shapefile that shows potential errors. **Distance** is the maximum distance (in meters) between landlocked and non-landlocked features that you want to consider potential errors; if the distance is set to 0, erroneous attribution will occur only when a landlocked feature physically touches a non-landlocked feature. Higher distance values will flag segments that potentially should connect to non-landlocked features but are within the specified distance away. For this example, run the tool with the distance set to 5.

The output will need to be visually inspected in order to determine whether attributes in the original data need to be changed and changes must be made manually.[[17]](#footnote-17) Imagery and/or digital elevation data can be used to inform this process. Three features were flagged as errors in the sample dataset. Two of these features clearly should have the value of the **LANDLOCK** field changed to 0. The third feature is several meters away from the nearest flowline. Underlying imagery may help determine whether there is a connection between the two features.

*Check waterbody snapping*

We developed a tool to determine whether stream lines were correctly snapped to the edges of lakes that they were associated with. This is necessary so that flowlines under lakes are not incorrectly attributed with their own distinct REACHID and so that flowlines not under lakes are not incorrectly attributed as part of lakes. Only relatively large difference (perhaps greater than a few meters) will make a difference in the output; however, this tool will output all possible errors. Incorrect snapping will only affect waterbodies used as lake seeds in the **Seed Generation** tool, so only lakes need to be checked by this tool.

Set the **Temporary Work Folder** to any location where you want temporary files to be stored; these can be deleted after the tool runs. Set **Input Waterbodies** to **/appendix/lakes\_fix.shp** and **Input Flowlines** to **/appendix/inputFlowlines\_fix.shp. Output Snap Errors** is the location and name of the output polygon that will contain errors. When you run the sample data, three line segments are output. All of these lines should be split and snapped to the edge of the lake polygon before proceeding with zone creation.[[18]](#footnote-18)

*Identify waterbodies that need lines*

Lakes that are connected to a stream network should have internal line segments, even if they are at the edge of the stream network. This ensures that the lakes will be attributed with proper to-from relationship data.

Set the **Temporary Work Folder** to any location where you want temporary files to be stored. Set **Input Waterbodies** to **/appendixData/lakes\_fix.shp** and **Input Streams** to **/appendixData /inputFlowlines\_fix.shp.** This feature will have two outputs: the **Output Error File** with waterbodies that should have lines added to them and the **Output Networked Waterbodies** file that returns all waterbodies that are intersected by flowlines. Waterbodies the user-specified **Distance** from stream lines will be considered networked but missing lines. The distance is in meters. Last, users need to specify the field (which cannot exist already in the waterbody shapefile) that will be used to indicate whether a feature in the **Output Error File** was erroneous because 1) it has a stream line touching but not crossing through waterbody (error code 1) or 2) it is a waterbody adjacent to a networked waterbody (error code 2).

Running the sample script outputs three features in the **Output Error File**. Two features are error type 1, meaning that they are touched by a stream flowline but the flowline does not cross through the feature. The third feature is error type 2, because it is adjacent to a networked lake but does not touch a networked line itself. The **Output Networked Waterbodies** file contains four waterbodies- those that are directly on a network and crossed by internal flowlines.

*Fix watershed expands*

The watershed delineation tool is designed to create watersheds that conform to the boundary of the input catchment features. In some cases, however, there are small gaps in some of the output watersheds. This means that there is some land in the study area that is not associated with any watershed. This tool will help a user fix these gaps. The file **/appendixData/watersheds\_fixExpand.shp** will be used for this tutorial; we manually added gaps to these watersheds because the actual watershed delineation process in this area did not lead to gaps.

The first step of the process is identifying whether there are any gaps in the output data. We will do this by dissolving the **watersheds\_fixExpand.shp** by the HUC12 identifier, then determining whether any of the dissolved features are not identical to the original HUC12 file. Open **Data Management Tools 🡺 Generalization -🡺 Dissolve**, set **watersheds\_fixExpand.shp** as the **Input Features**, set the output feature class to any location you select, and use the HUC12 field as the **Dissolve\_Field.** Run the tool with the rest of the arguments set to their default. Next, **Select By Location** , selecting features from **/inputs/HUC12.shp** that are identical to the features in the dissolved watershed file you just created. Open the attribute table for **HUC12.shp**, switch the selection, and export this file as a new shapefile with the name **hucs\_withGaps.shp**

Now we are ready to run the **Fix watershed gaps** tool. Set the **Input larger catchment boundary** as the **hucs\_withGaps.shp** file that was just created; the script will loop through each of these features to fix gaps in them.[[19]](#footnote-19) The **Unique identifier** can be set to the field **HUC\_12.** Set **Delineated Watersheds** to the previously created file **/outputs/watersheds.shp.** These watersheds will be used to inform the gap fixing process. The DEM can be set to **/inputs/DEM.img** and the **Output Fixed Watershed** will be the name and location of the output shapefile of watersheds with gaps fixed.[[20]](#footnote-20)

*Using connectivity scripts with current Hydro24kAttribution database*

The Hydro24k attribution data are set up to run connectivity traces without having to create a new landscape network. Following are instructions for running a trace script to find the distance to the nearest lake using the Hydro24k attribution data. With some modification, any of the trace attribution scripts can be run for the Hydro24k data.

First, from **Spatial24kHydro**,add the flowlines and relationships files to an open ArcMap document. Next, from **Hydro24k\_DATE.mdb**, add the **BaseAttributes\_24k** table. Join flowlines with **BaseAttributes\_24k** based on the **REACHID** field. Export the joined flowlines shapefile into a new shapefile. Add a short integer field called **StopTrace** to the exported shapefile. Select features where the field *Status* =”Great Lakes.” Update the **StopTrace** field to 1 for these selected features; all other features should have a 0 in this column. Clear the selection. Open the attribute table of the shapefile and export the table as a csv file (select data type “text file” but change the extension to .csv). Also export the relationships table to a csv file.

In the **~HydroAttribution -> Attribution -> 4 Nearest Lake** tool, set up the **R Directory, Connectivity script location, Maximum search distance, Lake Minimum Area,** and **Lake Maximum Area** as explained in the documentation in the main body of the tutorial. Add the newly exported relationship table to **Relationship Table** and set **FROM\_TRACEID** and **TO\_TRACEID** as the **From ID column** and **To ID Column.** Add the newly created csv flowlines file to **Flowline Data Table.** Set **Feature ID column** to the field **TRACEID***,* **Reach Field column** to **REACHID***,* **Length Field** to **Shape\_Leng** (or may be **Shape\_Length** if file was exporting directly from a geodatabase), **Dam Field** to **DAMSIDE**, **Stop Trace Field** to the new **StopTrace** field, and **Lake Area Field** to **Lake\_Area**.

1. These tools often require some massaging to work in ArcGIS, particularly versions of ArcGIS later than 9.2 and 9.3. The most important element is installing **pywin32** which can be downloaded at <http://sourceforge.net/projects/pywin32/>. Additionally, you may need to change the python command, gp = arcgisscripting.create() to gp = arcgisscripting.create(9.2) where 9.2 denotes your version of ArcGIS Desktop. If you have trouble getting the tools to run, refer to their documentation included in the **documentation** folder, or contact one of the three authors of this document. The tool also may not be able to create networks for more than approximately 100,000 features. [↑](#footnote-ref-1)
2. This tool may store temporary shapefiles called out\_edge.shp, outlet\_nodes.shp, and source\_nodes.shp in the users C:\TEMP directory. These files will need to be deleted between uses of the tool. [↑](#footnote-ref-2)
3. To help diagnose topological problems, use the “arrow at end” symbology on the **edges** feature class in your landscape network. If flow direction is correct, the topological error is likely due to disconnected features. [↑](#footnote-ref-3)
4. The output of this field may not be correct under all of the **Stream Selection Field** scenarios listed above [↑](#footnote-ref-4)
5. The tool edits out all flowlines that are not necessary to breach the HUC12 boundary. Waterbodies edited out by the tool can be inspected for potential issues in the file **/outputs/excludedWB.shp.** In many cases, this will be an empty shapefile. [↑](#footnote-ref-5)
6. There should be no more than 1000 unique seed IDs within any given redelineation unit (e.g., HUC12). If there are more than 1000, this script will crash due to a limitation in Spatial Analyst. [↑](#footnote-ref-6)
7. Vertices help to resolve competing watersheds at confluences and eliminate spurs from the output. Spurs are small parts of watersheds (or potentially other features, such as riparian buffers) that cross a confluence and impose into a connecting feature’s watershed. [↑](#footnote-ref-7)
8. For more information describing the process of DEM conditioning, refer to the Arc Hydro Tools Tutorial at http://www.crwr.utexas.edu/gis/gishydro06/ArcHydro/ArcHydroTools/Doc/Arc%20Hydro%20Tools%201.1%20-%20Tutorial.pdf [↑](#footnote-ref-8)
9. As in the **Seed Creation** and **Delineate Watersheds** tools, the **Create Riparian Buffers** tool is hard-coded to ignore values associated with Great Lakes features (i.e., features that are *within* the boundary of a Great Lake). These special values are **600000001** and **600124406**. [↑](#footnote-ref-9)
10. It is likely that the tool will crash for very large study scopes (e.g., the entire state of Wisconsin at 10-meter resolution). Start small. [↑](#footnote-ref-10)
11. Prior to running the tool, you must run a complete (no Z limit, and thus, no sinks) “fill” on the DEM. The **Fill** tool can be found in **ArcToolbox🡺Spatial Analyst Tools🡺Hydrology**. [↑](#footnote-ref-11)
12. The downstream feature (value of **TOCATCHID** or the “to” in “to/from” topology) is identified by selecting the neighboring feature that has the highest flow accumulation (derived from flow direction raster) on a bordering pixel. This pixel denotes the theoretical “spill point” of the watershed. [↑](#footnote-ref-12)
13. If any of the input attribute rasters do not match the zonal raster in resolution, cell size, or projection, an output file of errors will be saved in the output location instead, indicating that some of the attribute rasters may need to be resampled and/or reprojected. [↑](#footnote-ref-13)
14. Features with downstream dams will only be connected to upstream features and features with upstream dams will only be connected to downstream features. [↑](#footnote-ref-14)
15. Only one of the two lakes in **/outputs/lakes.shp** ends up being associated with the **edges** file. This occurs because the other lake is not connected to the network. [↑](#footnote-ref-15)
16. Output in this field may be somewhat buggy and should be used with caution. The field can provide a general idea of whether features did not reach lakes because there were no lakes within the set distance or whether the features were hemmed in by dams, but the output may not be exact. [↑](#footnote-ref-16)
17. Changes may need to be made to the output features, features near the output features, or a combination of the two. [↑](#footnote-ref-17)
18. If you split line segments, do not forget to assigned each line segment a unique identifier. [↑](#footnote-ref-18)
19. This tool makes use of the Cost Allocation function in ArcGIS, which is somewhat computationally intensive and may on occasion cause ArcGIS to crash. If you have trouble getting the script to run, you can input each boundary individually and then use the output watersheds as the input delineated watersheds when you run the tool for the next boundary. [↑](#footnote-ref-19)
20. Users may want to check the accuracy of the output file by doing another dissolve and checking that features are identical to input HUC12 features. [↑](#footnote-ref-20)