Construction of a gis_travel_times_master.db database that supports haul cost calculation in BioSum

Prepared by Drue Lindstrom, Demetrios Gatziolis and Jeremy S. Fried

1 March 2025

The material in this document will also be incorporated into the next version of the BioSum Users Guide when supplemented with context on the purpose, assumptions, limitations and interpretation considerations related to this database when used for a BioSum analysis. Here we specifically document and describe the process used to build what we call the "travel times database" (gis_travel_times_master.db dated 17 Sept 2024) provided with BioSum 5.11.x at http://biosum.info/downloads/spatial_data.html for the purpose of computing haul costs for merchantable and residue wood modeled as moving from the forest to processing facilities.

Most BioSum users in 12 western states (AZ, CA, CO, ID, MT, NM, NV, OR, SD [Black Hills region only], UT, WA, and WY) will not need to implement this ArcGIS and R based workflow because the database provided at the above link will contain wood using facilities relevant to the landscape they are analyzing. Some analysts may wish to include new or prospective processing sites not found in the BioSum-provided database, to account for changes in travel time in portions of a project area generated by influential new or upgraded roads or to utilize newly available road layers considered to be more complete or accurate than was available when the BioSum-provided database was constructed. If a new or prospective facility will be located at or very near an existing facility, there may be no need to follow this workflow as records can be added via database queries by generating modified copies of records in the existing database's processing_site and travel_time tables associated with the co-located existing facility (i.e., with the new facility's PSITE_ID, PSITE_CN, NAME and related information in the processing_site table and new PSITE_ID in the travel_time table). This "shortcut" workflow will be included in a future edition of the BioSum Users Guide.

If conducting BioSum analyses outside these 12 western states changes or needing to make changes more extensive than adding a new facility co-located with an existing one, this workflow, faithfully followed, will generate a replacement gis_travel_times_master SQLite database that can be copied to the appdata\Roaming\FIABioSum folder (after backing up the existing one) so that new travel times data can be loaded to a BioSum project (replacing any existing travel time data in that project). Users wishing to build their own travel times database may wish to begin with the plots and processing sites in the BioSum-provided version of this database.

This workflow ingests 3 GIS layers as inputs—roads, plots and processing sites. Sites may include currently operating processing facilities that accept (and pay for) merchantable wood or "dirty chips" feedstocks. They may also include locations that previously hosted such facilities or that are being considered for hosting a facility in the future. Road layers will need information to support the assignment of a speed rating to every road segment. Workflow output consists of a very large table of travel times from the nearest on-road point associated with each plot to each processing site, within a user defined search distance. BioSum uses this travel-time table to estimate the cost of hauling harvested wood.

There are two ArcGIS workflows and one workflow that relies on R scripts: 1) Preparing the plot and processing site data, 2) compositing one or more transportation layers (these are typically road layers, but rail travel can also be modeled), and 3) applying a series of R scripts, developed by PNW Research Station scientist Demetrios Gatziolis, to the outputs of the first two workflows to assess and address data quality/integrity, and ultimately generate the travel times table following an algorithm based on graph theory.

1. Preparing plot and processing site data

This workflow depends on plot and processing site data being in CSV (comma separated value) formatted text files. If these data exist in SQLite or Microsoft Access databases, they can be easily exported as CSV files as follows (steps accurate as of 2024 for SQLiteStudio v3.4.4 and MS Office 365 Access version 2501).

To export from SQLite to CSV:

1. In the Database panel, double click the table you want to export to open it
2. Click <Export Table> ()
3. Ensure the Database and Table parameters are set to the table you want to export
4. Deselect all Options except for Export table data

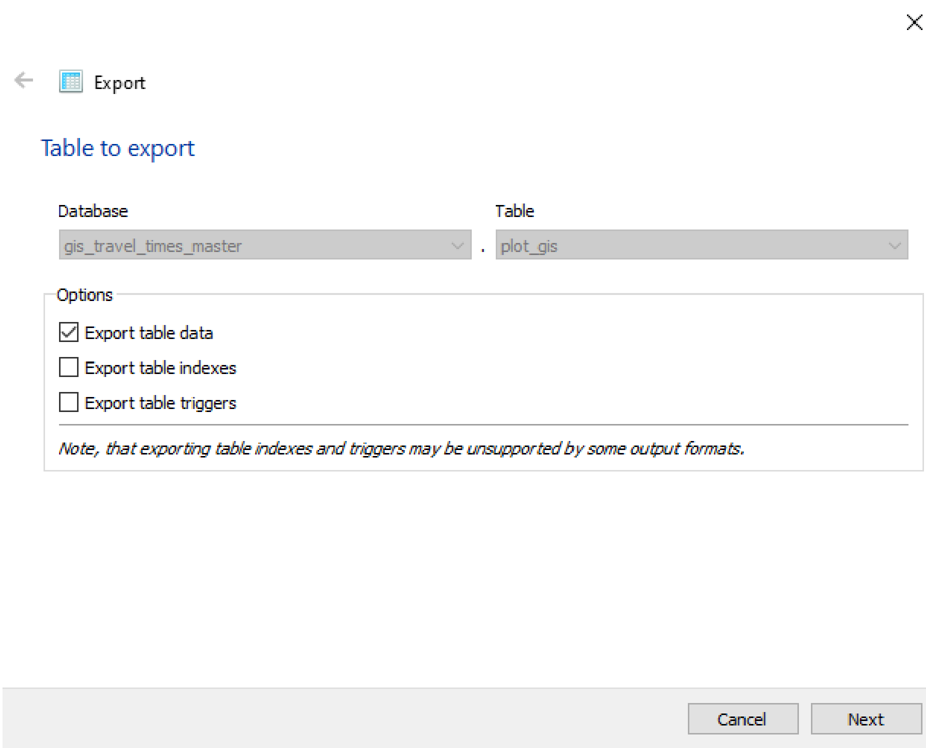


Figure #1 – SQLite Export window, Table to export screen

5. Click <Next>
6. In the Export format dropdown, select CSV
7. Select a destination and name for the output CSV file
8. In the Export format options area, select Column names in first row
9. In the Column separator dropdown, select “, (comma)”

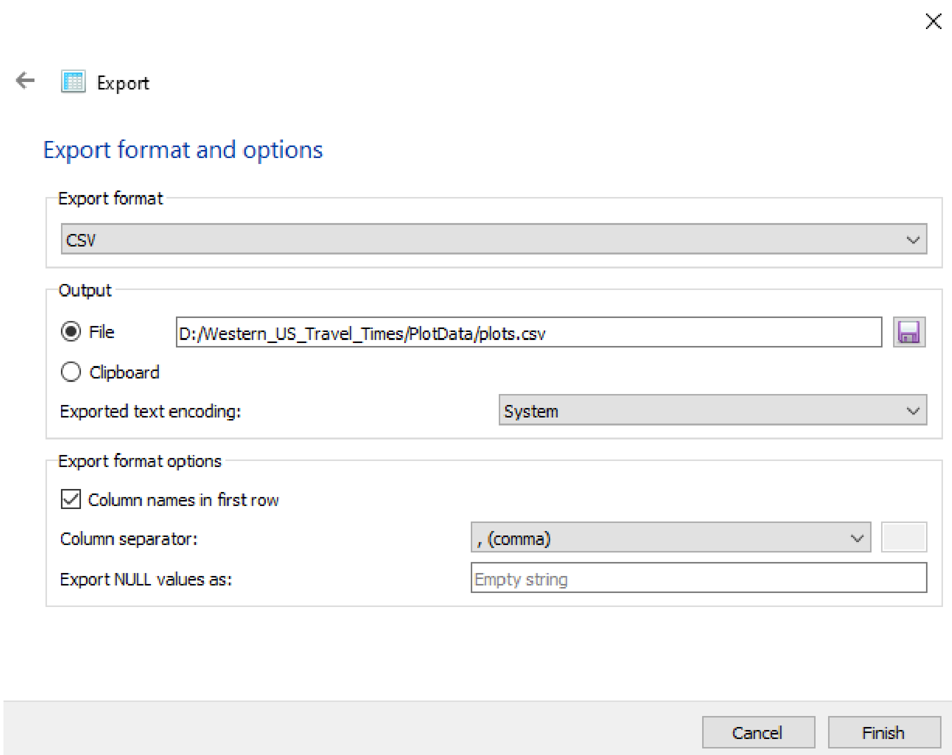


Figure #2 – SQLite Export window, Export format and options screen

10. Click <Finish>
11. Inspect the output CSV file to ensure it looks correct

To export from Microsoft Access to CSV:

1. In the All Access Objects pane, click the table you want to export to select it
2. On the External Data tab in the Export group, click <Excel> to open the Export - Excel Spreadsheet window
3. Set a destination for the output file
4. Use the File format dropdown to select Excel Workbook (*.xlsx)

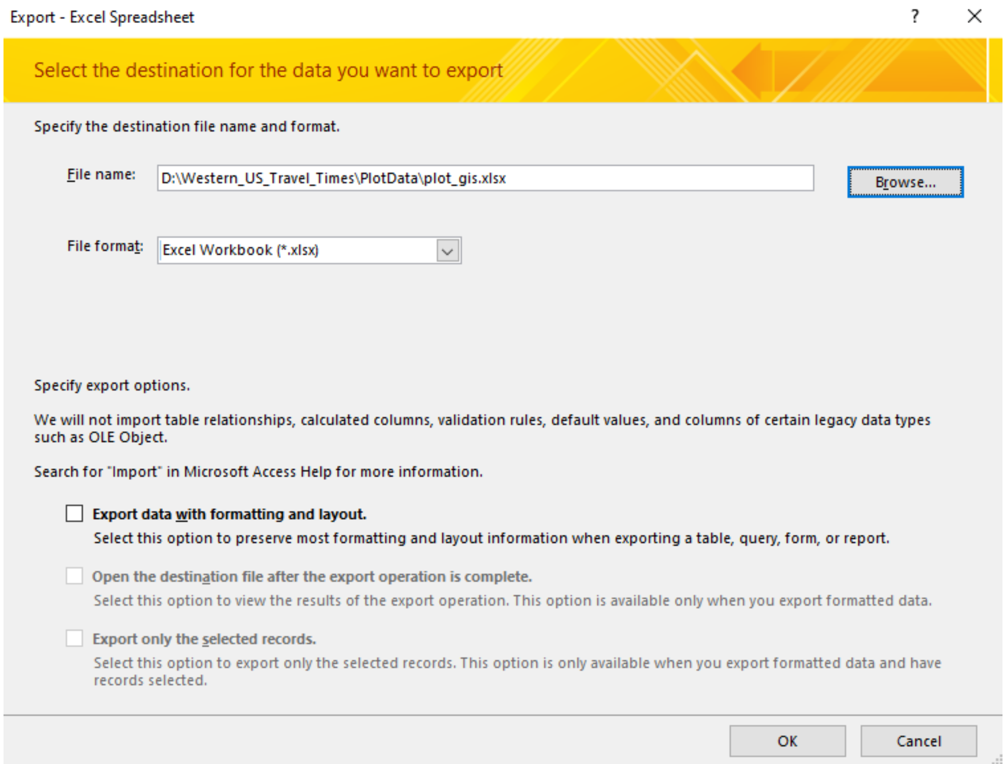


Figure #3 – Microsoft Access Export - Excel Spreadsheet window

5. Click <OK>
6. If the Save Export Steps window opens, you can ignore it and click <Close>
7. Open the output Excel worksheet
8. On the File tab click <Save As>
9. Use the file format dropdown to select CSV (Comma delimited) (*.csv)
10. Click <Save>

The travel times workflow requires that the plot data include the following fields (and data types) populated as specified.

Field Name	Data Type	Field Contents
STATECD	Integer	The FIPS code of the state the plot is in
COUNTYCD	Integer	The FIPS code of the county the plot is in

PLOT	Integer	The FIA plot number
PLOT_CN	Integer	7-digit combination of STATECD and PLOT. In Excel, use this formula to create the PLOT_CN: ***STATECD cell* * 100000 + *PLOT cell;* note**—this is unrelated to the PLT_CN column found in FIADB tables such as COND and TREE, which is formulated entirely differently
ORIG_LAT	Double	The latitude of the plot (in decimal degrees)
ORIG_LON	Double	The longitude of the plot (in decimal degrees)

Table #1 – Plot CSV schema

Processing site data must contain these fields, data types and contents:

Field Name	Data Type	Field Contents
PSITE_ID	Integer	Unique, incrementing integer starting at 1 to uniquely identify each processing site within a project
PSITE_CN	Text (character limit 12)	A character string that uniquely identifies processing sites across projects. This is a combination of the latitude, longitude, and biocd. In Excel, use this formula to create the PSITE_CN: TEXT(ABS(*LAT cell*)*1000, "00000") & TEXT(ABS(*LON cell*)*1000, "000000") & TEXT(*BIOCD cell*, 0)
NAME	Text (character limit 100)	Name of the processing site
TRANCD	Integer	1 = Regular (Processing Site Only Accessible by Road) 2 = Railhead (Transfer site of biomass from truck to rail)

		3 = Rail Collector (Processing Site Accessible By both Road and Rail)
TRANCD_DEF	Text (character limit 40)	“Regular”, “Railhead”, “Rail Collector”
BIOCD	Integer	1 = Merchantable 2 = Chips 3 = Both 4= Facility type TBD (placeholder for travel time calculations without commitment to a particular facility type; new type will be addressed in a future BioSum release)
BIOCD_DEF	Text (character limit 40)	“Merchantable”, “Chips”, “Both” “Placeholder”
EXISTS_YN	Text (character limit 1)	Has possible values of Y and N. All mills that are open and operational should be Y. All other mills should be N; note that either type can be selected for inclusion in a BioSum analysis
LAT	Double	The latitude of the processing site
LON	Double	The longitude of the processing site
STATE	Text (character limit 2)	The state the processing site is in

COUNTY	Text (character limit 40)	The county the processing site is in. It is possible for this field to be blank if the value is unknown
CITY	Text (character limit 40)	The city the processing site is in. It is possible for this field to be blank if the value is unknown
STATUS	Text (character limit 40)	Current status of the mill. Can be anything, but some helpful options are Operational, Closed, Idle, and Planned
MILL_TYPE	Text (character limit 40)	Type of mill. Some examples include sawmill, biomass/energy, post/pole, and plywood. The BIOCD should correspond to the mill type
NOTES	Text (character limit 100)	Any notes about the mill. This field can be blank

Table #2 – Processing site CSV schema

NOTE: It is very important for the data tables to be prepared as just described; any departure may result in having to repeat some or all of the processes in this workflow. If questions, please contact support@biosum.info.

Setting up a geoprocessing environment if using Arc GIS Pro

1. Open ArcGIS Pro and log in if needed
2. Under New Project, click <Map>
3. Give the project a name and set the location to your desired folder
4. Optionally, uncheck the Create a folder for this project check box if you don't want to store the project file in its own folder

The workflow outlined below involves several datasets from multiple sources. It is recommended to organize your data to make it easy to find. For example, keep the project file, plot CSV, and processing site CSV in a project folder and road datasets in another folder within that folder.

Creating a point feature class from plot coordinates

1. Open the XY Data to Point tool in the Data Management toolbox by navigating the Geoprocessing pane or by selecting <Add Data><XY Point Data> in the Layer group of the Map tab
2. Set the Input Table parameter to the CSV containing the plot data by clicking the folder to the right of the parameter input and navigating to the CSV
3. Name the output something logical and easy to find
4. The X Field and Y Field parameters should automatically populate. If not, use the drop downs to select the X Field to LON and the Y Field to LAT
5. Set the Coordinate System parameter to the correct coordinate system for the data source (most plot data from the FIA Datamart uses the NAD 83 datum). Click the globe to the right of the parameter input to search for the correct coordinate system
6. Click <OK> to run the tool

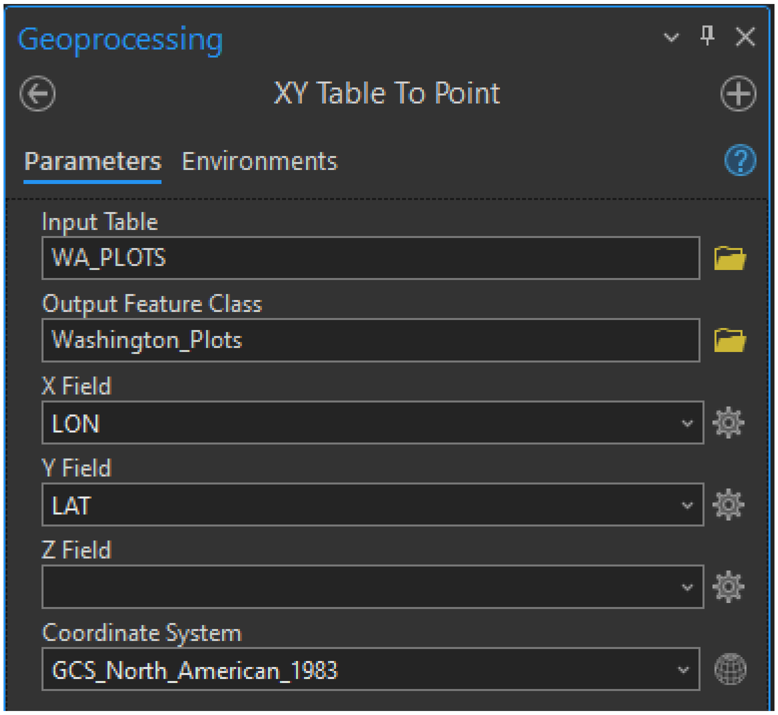


Figure #4 – ArcGIS XY Table To Point tool

NOTE: If the points do not show up where expected, the Coordinate System is probably incorrect. Double check the coordinate system in the metadata of your plot data source

Using GIS to subset plots to the desired study area

1. Display the point feature class created in the above workflow. The plots can be subset based on a spatial selection or from attributes.
   a. Subsetting based on attributes
      i. On the Map tab in the Selection group, click <Select by Attributes> ( )
      ii. Set the Input Rows parameter to the plot layer
      iii. Choose values from the drop-down menus to build a where clause
      iv. Ensure the Selection Type parameter is set to New Selection
      v. Click <OK> to run the tool
   b. Subsetting based on location

    i. Add an existing polygon representing the study area to the project or manually create one
    ii. On the Map tab in the Selection group, click <Select by Location> ()
    iii. Set the Input Rows parameter to the plot layer
    iv. Set the Selecting Feature parameter to the study area polygon
    v. Choose a Relationship. It should usually be set to Within
    vi. Ensure the Selection Type parameter is set to New Selection
    vii. Click <OK> to run the tool

2. With the desired subset of plots selected, open the Feature Class to Shapefile tool in the Conversion toolbox. Ensure the tool notifies you that the input has a selection
3. Click <OK> to run the tool

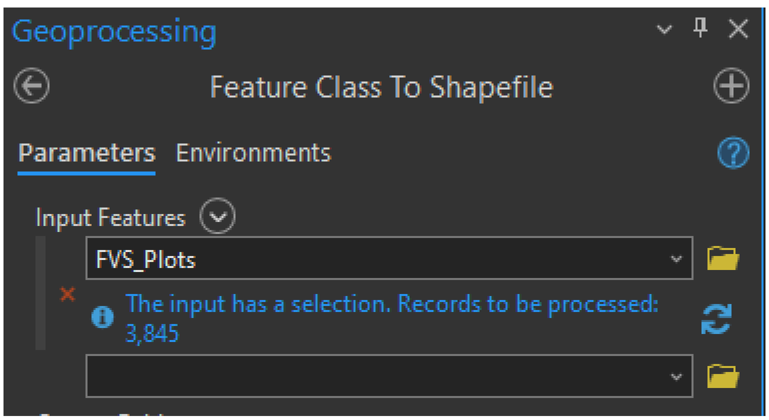


Figure #5 – ArcGIS Feature Class to Shapefile tool

4. Add the new shapefile to your project. This is the plot layer you will use for the rest of the workflow

Creating a processing site point feature class

You will follow the same steps used to create the plot point feature class.

1. Open the XY Data to Point tool in the Data Management toolbox by navigating the Geoprocessing pane or by selecting <Add Data><XY Point Data> in the Layer group of the Map tab

2. Set the Input Table parameter to the CSV containing the processing site data by clicking the folder to the right of the parameter input and navigating to the CSV
3. Name the output something logical and easy to find
4. The X Field and Y Field parameters should automatically populate. If not, use the drop downs to select the X Field to LON and the Y Field to LAT
5. Set the Coordinate System parameter to the correct coordinate system for the data source. Click the globe to the right of the parameter input to search for the correct coordinate system
6. Click <OK> to run the tool

2. Preparing transportation layers

Road data needs to be comprehensive (contain all extant roads and with luck, no non-roads such as trails or rights of way that may also present as linear features) and contain attribute(s) that can be used to reasonably infer the likely speed at which logging trucks would travel. This could be a speed limit or an adaptation of such (e.g., as a percentage or capped). Some common useful attributes are road type and functional class. To achieve a comprehensive dataset with major roads, minor roads, rural roads, and forest roads, multiple datasets for each state within the study area are typically required. Some datasets may have a speed attribute. If you plan on using an existing speed attribute, spot check some cases to ensure credible values that are evenly divisible by 5. The table below lists data sources used to create the gis_travel_times_master database. If you need to look for additional data, a good place to start is by looking for an open spatial data portal for the state(s) you need data for.

State/Region	**Data Source**	**Name of dataset used in gis_travel_times_master.db**
Arizona	AZGeo Data	AZ All Roads Network 2021
California	California State Geoportal	CRS - Functional Classification
Colorado	Colorado Geospatial Portal	Highways

		Major Roads
Idaho	Idaho State University - MILES	Idaho Basic Features geodatabase
Montana	Montana State Library	Transportation Shapefile or geodatabase
Nebraska	NebraskaMap	Street Centerlines
Nevada	NDOT GeoHub	Statewide Routes
North Dakota	North Dakota GIS Hub Data Portal	NDGISHUB Census Bureau TIGER Roads
Oregon	Oregon Department of Forestry ArcGIS Hub	Transportation ODF Statewide Road
South Dakota	South Dakota GIS Data	Local Roads
Utah	Utah SGID	Utah Roads
Washington	Washington DNR GIS Open Data	WADNR Active Roads
	WSDOT Geospatial Open Data Portal	WSDOT - State Route Lines (1:24K) Current
Wyoming	USGS Data Series 821	Wyoming Roads 2009 geodatabase
National	OpenStreetMap	Several .shp.zip files
	U.S. Forest Service - Geospatial Data Discovery	National Forest System Roads (Feature Layer)

Figure #3 – Suggested road layer data sources

1. Add road datasets to your ArcGIS Pro project.
2. Remove roads of a type likely to be impassable by truck. For example, some "road" datasets are really transportation layers that also contain bike trails and/or pedestrian paths.

    a. On the Map tab in the Selection group, click <Select by Attributes> ()

    b. Set the Input Rows parameter to the road layer you are working on

    c. Choose values from the dropdown menus create a where clause that selects road segments to be deleted

    d. Ensure the Selection Type parameter is set to New Selection

    e. Click <OK> to run the tool

    f. On the Edit tab in the Features group, click <Delete>

    g. On the Edit tab in the Manage Edits group, click <Save>

3. Right click on the road layer in the Contents pane and choose <Open Attributes Table>

4. At the top of the Attribute Table pane in the Field group, click <Add> () to create a new field

5. Name the new field “MAX_MPH” and set the Field Type to Short

6. Click <Save> in the Changes group of the Fields tab

7. Close the field editing pane

8. Right click the new MAX_MPH field and select <Calculate Field> ()

9. In the box under MAX_MPH =, enter “calcmaxmph(“ and double click the field name in the list above. The field name should populate after the opening parenthesis and enclosed in exclamation points. If you want to use multiple fields, separate them by commas. Enter a closing parenthesis into the box. It should look like **calcmaxmph(!fieldname!)**

10. In the Code Block box, enter “def calcmaxmph(“, then create and enter a variable name for the input field(s), and end by entering “):”

11. On the next line, start building your logic used to set the MAX_MPH based on different field values. A simple example using road type is shown below. The values will likely be different for you, but the structure should remain the same

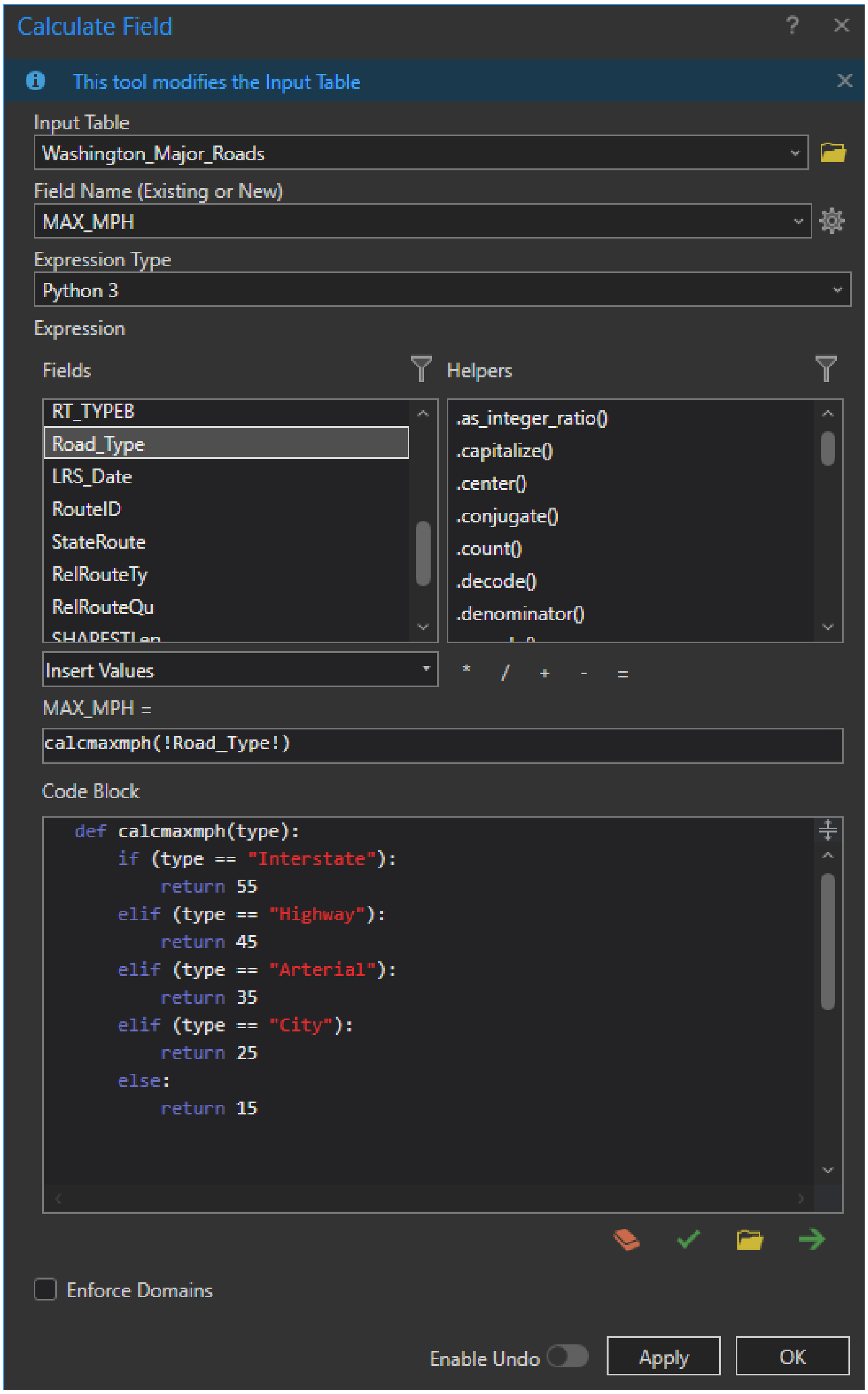


Figure #6 – Sample MAX_MPH calculation input

12. Click <OK> to run the tool and populate the MAX_MPH field
13. Repeat steps 2 - 12 for each road dataset

14. When multiple road datasets are needed, they must be merged into a single layer after the MAX_MPH field has been created and populated in each:
    a. Open the Merge tool in the Data Management toolbox.
    b. Set the Input Datasets parameter to include all roads layers
    c. Name the output something logical and easy to find
    d. In the Field Map parameter, all fields except for the MAX_MPH can be removed. Set the Merge Rule for MAX_MPH to Maximum
    e. Click <OK> to run the tool
15. Next, the roads can be clipped to only include a relevant buffer around the study area and included processing sites. Open the Buffer tool in the Analysis toolbox
16. Set the Input Features parameter to the processing sites layer
17. Name the output something logical and easy to find
18. Set the Distance parameter to 100 miles or another value thought to likely result in no unintended loss of connectivity in major roads that lead to processing sites
19. Leave the Method parameter as Planar
20. Set the Dissolve Type parameter to "Dissolve all output features into a single feature"
21. Click <OK> to run the tool

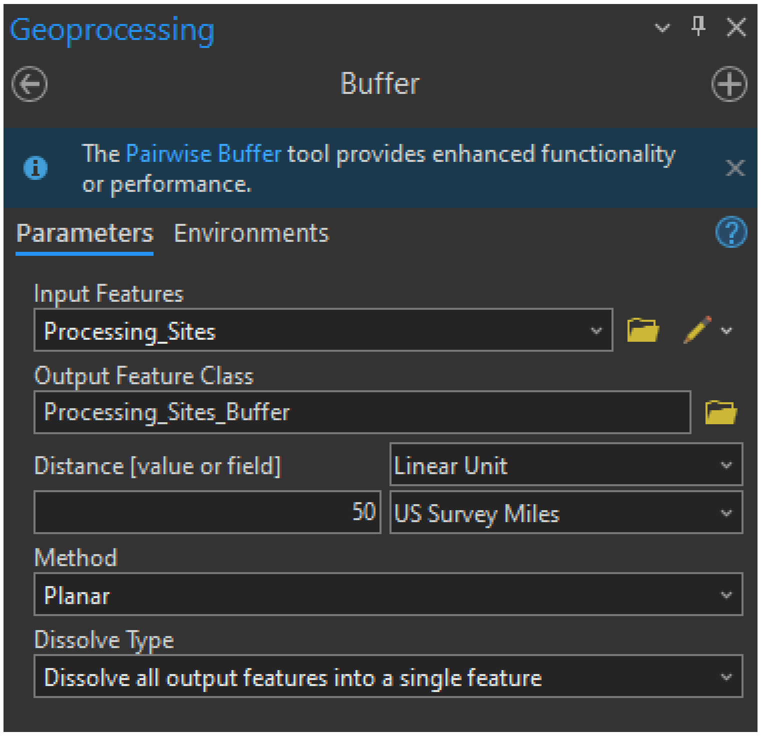


Figure #7 – ArcGIS Buffer tool

22. Visually inspect the output buffer polygon. It is possible that it has holes, does not fully encompass all plots, or does not fully encompass all major roads that would be reasonable travel routes. If any of these are the case, the buffer polygon can be manually edited. The shape of the buffer polygon does not need to be perfect as long as it encompasses all features it needs to
    a. On the Edit tab in the Tools group, select <Edit Vertices> ( )
    b. Click on the buffer polygon
    c. Use the Add and Delete tools in the editing toolbar to add and delete vertices as needed
    d. Click, hold, and drag vertices to move them

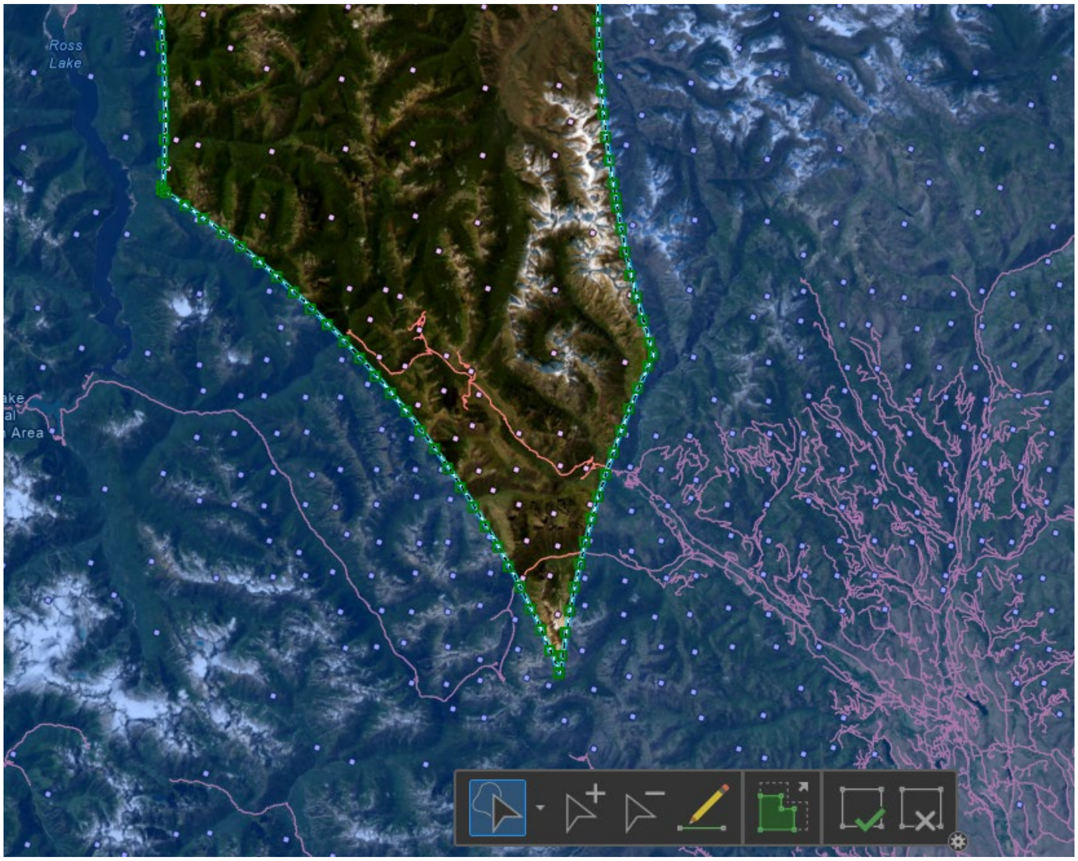


Figure #8 – Sample of buffer polygon that needs to be fixed

   e. When you have finished editing the polygon, click <Finish> ( ) in the editing toolbar

   f. On the Edit tab in the Manage Edits group, click <Save>

23. Open the Clip tool in the Analysis toolbox
24. Set the Input Features parameter to the merge road layer
25. Set the Clip Features parameter to the buffer polygon
26. Name the output something logical and easy to find
27. Click <OK> to run the tool
28. Reproject the output clipped road layer if it is not in the same coordinate system as the plots and processing sites

   a. Open the Project tool in the Data Management toolbox

   b. Set the clipped roads layer as the Input Dataset

c. Name the output something logical and easy to find
d. Use the dropdown menu to set the Output Coordinate System to the same as the plot or processing site layer
e. Click <OK> to run the tool

29. Open the Feature Class to Shapefile tool in the Conversion toolbox
30. Set the input to the clipped road layer with the correct projection
31. Click <OK> to run the tool

Congratulations—your data is now prepared for travel times processing!

3. Travel time processing with transportation_cost.R

The calculation of travel times is carried out by executing commands in the R script Transportation_cost.R which calls functions in BIOSUM_functions.R or BIOSUM_funcions.cpp, all of which were designed and coded by USDA Forest Service scientist Dr. Demetrios Gatziolis specifically for this workflow. They make use of graph theory applied to with each vertex of the road network considered a node and each network segment an edge. By allowing for multiple edges linked to a node, this construct accommodates road intersections including those collocated with the presumed forest landing locations and processing facilities). This script-based workflow requires only a rudimentary understanding of R (enough to edit file and path names in a script, and to execute code in chunks in the R or Rstudio environment, for example) and is outlined in this flow chart:

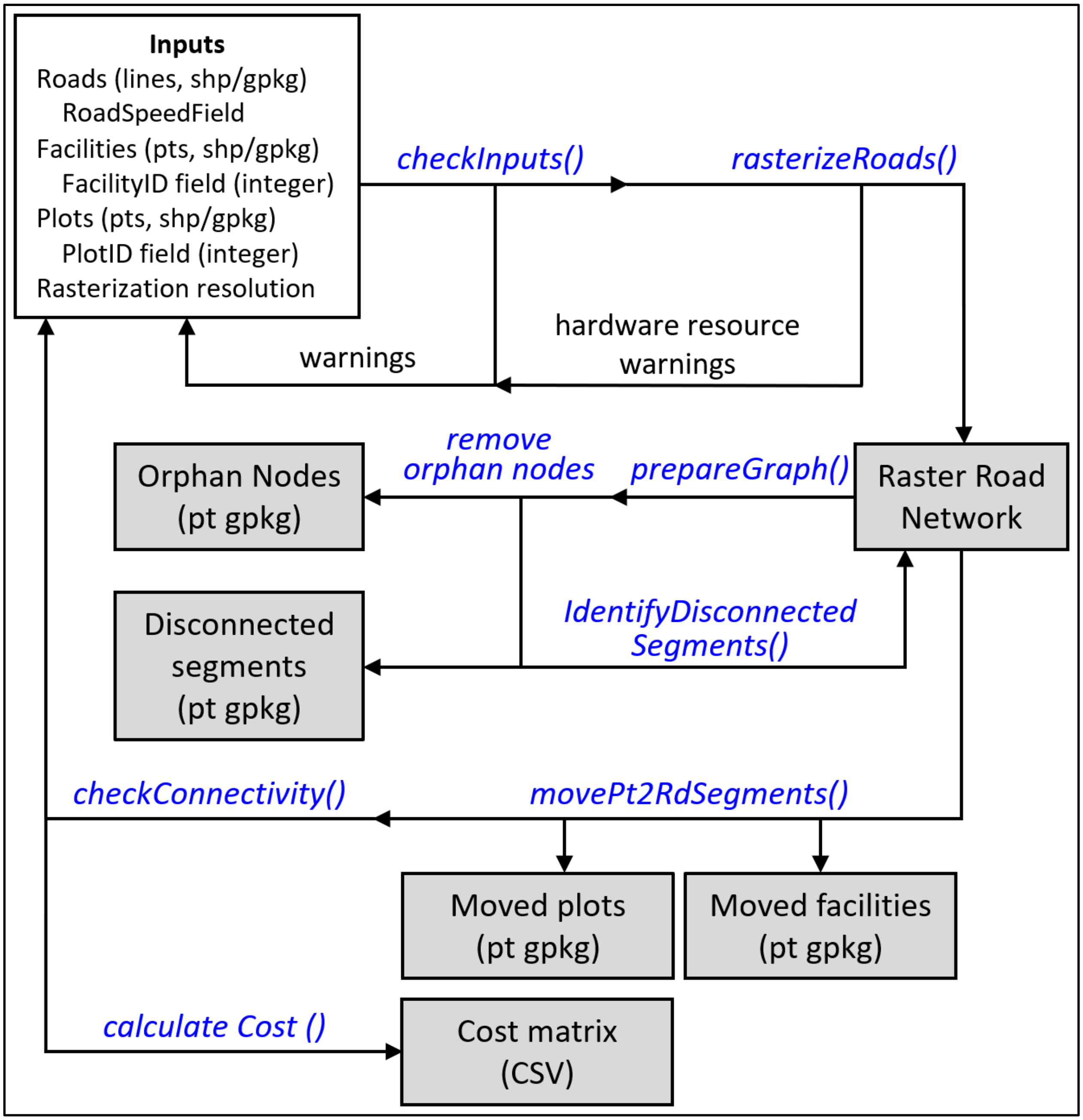


Image #9 – Visualization of the Transportation_cost.R script workflow

The scripts make use of six packages: terra, sf, doSNOW, cppRouting, RcppParallel, and, optionally, Rcpp. The packages are collections of functions that in other programming environments are typically known as libraries or modules. If not already present, each package must be installed in the R environment on the computer where this processing takes place. Package installation can be performed from the R command line using the following syntax: **`install.packages("NameOfPackage")`**, for example **`install.packages("sf")`**. For this workflow, all required packages can be installed at one step issuing **install.packages(c("terra", "sf", "doSNOW", "cppRouting", "RcppParallel", "Rcpp")**. Packages can also be installed using the R GUI as described below.

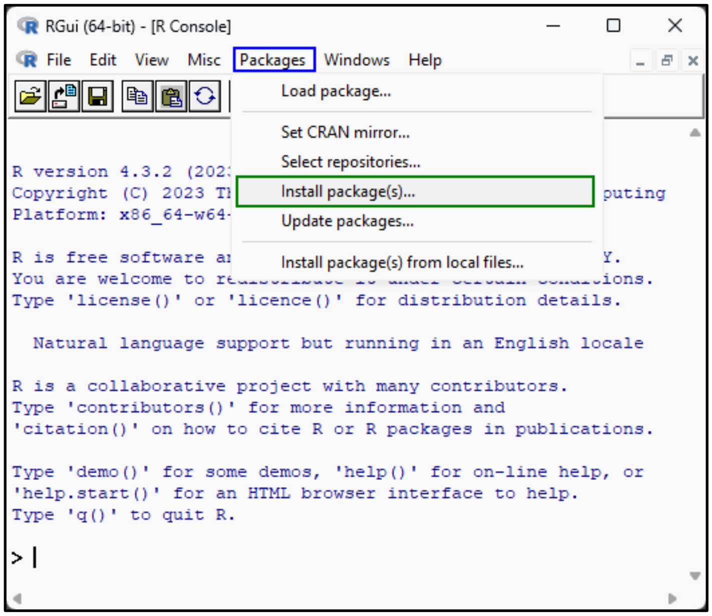


Image #10 – Location of "Install package(s)…" window

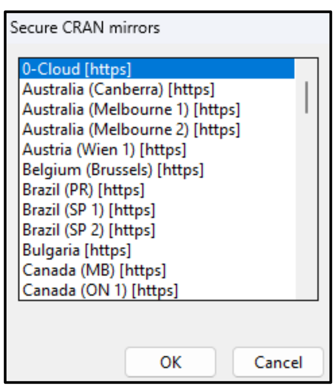


Image #11 – Select the CRAN mirror closest to your location

Using the slider to the right of the Packages window, select the packages needed. Holding down the control keyboard key allows for multiple package selections.

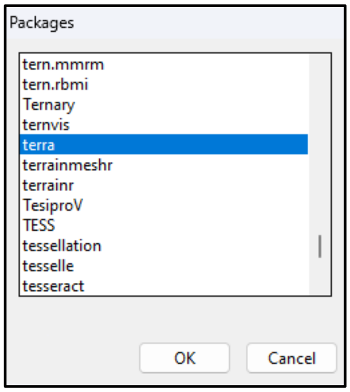


Image #12 – Packages window. Select the package(s) you want to install

Package installation requires network connectivity. Without a network connection, installation can make use of compressed package archives, in tar.gz format, obtained by alternate means. An option would be to use a networked computer, save the compressed archive file on a portable drive, and copy it to the computer that lacks network connectivity. The installation uses the following syntax: `install.packages(path_to_file, repos = NULL, type="source")`. The following example details the off-network installation of package terra:

```
url <- "https://cran.r-project.org/src/contrib/Archive/terra/terra_1.8-15.tar.gz"
```

```
download.file(url, destfile = "terra_1.8-15.tar.gz")  # from a networked computer
```

```
install.packages("terra_1.8-15.tar.gz", repos = NULL, type = "source")
# after terra_1.8-15.tar.gz is copied locally
```

Package installation from source does require Rtools.

The size of the graph that these scripts construct and operate on requires substantial amounts of available RAM. As much as 128 GB, or more, may be needed for a state the size of California if a fine cell size (e.g., 10m) is selected, though with a cell size of 30 or 40m, even 16 GB may be adequate for not-too-large landscapes.

The spatial data inputs developed in the ArcGIS Pro workflows described above (roads, plots, facilities) can be supplied as geopackages or shapefiles. Shapefiles are the recommended choice for roads. The code below can be run, one line at a time at the R or RStudio console/command line, to convert between shapefiles and geopackages (assuming the packages listed above are already installed in your R configuration):

Shapefile to geopackage

```
library(terra)
arcObj = vect("shapefile.shp")
writeVector(arcObj, "geopackage.gpkg")
```

Geopackage to shapefile

```
library(terra)
arcObj = vect("geopackage.gpkg")
writeVector(arcObj, "shapefile.shp")
```

The user must also specify values for the variables listed in Figure 13 and described below.

`rdName` is the name of the file that contains the road data, either as a shapefile or a geopackage. If the file is not in the working directory, the full path and name should be specified. The same applies to variables `facilityName` and `resourceName`. `rdField`, `facilityField`, and `resourceField`, denote corresponding, case-sensitive field (or column) names for each input dataset. The road segment values of variable `rdField` should be quantified as miles per hour. The values in `facilityField` and `resourceField` must be unique (no duplicates).

Variable `useCompiledCode` is of type logical and accepts only TRUE or FALSE values. If set to TRUE, the scripts use functions in BIOSUM_functions.cpp that are coded in C++ and offer improved computational efficiency compared to corresponding functions in BIOSUM_functions.R used when `useCompiledCode` is set to FALSE. However, use of C++ code embedded in BIOSUM_functions.cpp functions requires prior compilation that must be repeated in every new R or RStudio session. C++ code compilation is performed by issuing **`Rcpp::sourceCpp("BIOSUM_functions.cpp")`** and requires that Rtools are installed on the computer and their bin directory is on the operating system's path. As before, if BIOSUM_functions.cpp file is located outside the working directory, the path to the file must be specified in the sourceCpp function. Instructions on how to install the current version of Rtools (4.4) are available at https://cran.r-project.org/bin/windows/Rtools/rtools44/rtools.html. Adding the path to a computer running Windows involves typing **`env`** on the Search tab, selecting 'Edit the system environment variables', locating 'path' in the 'Edit environment variable' and clicking 'edit'. In the user interface that is activated, select 'New' and add the path. Assuming that Rtools are installed in the default location, the path for the current version would be `c:\rtools44\usr\bin`. Owing to the complexity of the process needed to enable C++ functionality in the workflow, and the requirement for elevated computer management privileges to perform the Rtools installation, the default value of `useCompiledCode` is set to FALSE.

The value for the `rasterResolution` variable specifies the resolution of the rasterized road dataset in meters. For example, a value of 50 will generate a binary road raster with 50 meter cells with each cell representing "road" or "not road."

This workflow requires that all nodes (cells labeled 'road') of the raster road representation are connected, that is, any destination node can be reached starting from any other start node. Testing connectivity between all pairwise node combinations for a road network of modest size, say 10,000,000 nodes, is practically infeasible. For various reasons, a raster road network generated from a vector road representation will initially (before any editing) contain numerous instances of disconnected segments each comprising one to many cells. Testing connectivity using a randomly selected pair of 'from' to 'to' nodes (cells), instead of all possible combinations, will often involve nodes that belong to isolated segments. The connectivity results of such a test will be erroneous. The probability of such an error decreases rapidly as we increase the number of 'from' to 'to' pairs we use in the test. The value of the `nIterations` variable specifies the number of such node pairs that will be used to check for network connectivity. Higher values support more trustworthy connectivity test results. We have determined empirically that a value of 10 suffices in almost all cases. The `nIterations` variable is used in conjunction with the `connectedNodeThreshold` variable. The latter is the minimum ratio of nodes that the largest identified connected (sub)network must have compared to all nodes. The default value of 0.75 expects at least 75% of all nodes to be connected in a single subnetwork. If the largest connected subnetwork the process identifies fails to reach the `connectedNodeThreshold` value after `nIterations`, an error is returned. In such a case, the input vector representation of the road network likely contains serious connectivity deficiencies, except perhaps in the presence of a large subnetwork on an island or other legitimate areal subset.

The transportation cost R script outputs 6 files: a raster version of the input road data, a geopackage of the processing sites moved to the nearest road, a geopackage of the plots moved to the nearest road, a text file in CSV format with the travel times in hours between each plot and each processing facility, a geopackage of road segments that are disconnected from the road network, and a geopackage of road segments that are disconnected from the road network, and a geopackage of orphaned road segments. Each of the disconnected segments contains at least two nodes. Orphaned road segments contain only one isolated node (single cell) and usually are undetected remnants of a topological operation performed on the vector representation of the road network. The user can specify the output folders and names for the road raster, moved facility, moved plots, and cost CSV files. The disconnected and orphaned road segments geopackages are automatically added to the work directory of the active R session.

1. Before running the script, ensure that the BIOSUM_functions.R file is in the session work directory. If `useCompiledCode` is set to TRUE, the BIOSUM_functions.cpp file should also be in the work directory. Then enter **`setwd("folderpath")`** in the R or RStudio console/command line
2. Close out of any other software that has access to <u>any</u> files used in the script, **especially ArcGIS –** known for a tendency to indiscriminately apply locks to files it accesses
3. Update the `rdName` parameter to your road shapefile. If the road shapefile is in a different folder than your work directory, be sure to include the file path
4. Update the rdField parameter to the name of your speed field if it is different than the default MAX_MPH
5. Update the `facilityName` parameter to your processing sites shapefile or geopackage. If the processing site file is in a different folder than your work directory, be sure to include the file path
6. Update the `facilityField` parameter to the name of your unique processing site ID field if it is different than the default PSITE_ID
7. Update the `resourceName` parameter to your plot shapefile or geopackage. If the plot file is in a different folder than your work directory, be sure to include the file path
8. Update the `resourceField` to the name of your unique plot ID field if it is different than the default PLOT_CN
9. Change the `useCompiledCode` parameter to TRUE if desired
10. If desired, change the rasterResolution parameter. It is recommended not to use a value above 50 because of ensuing unwarranted generalization of the road network that enforces connectivity between disjointed network segments, as, for example, between two roads on opposite sides of a river without a bridge connecting them.
11. If desired, change the `nIterations` and `connectedNodeThreshold` parameters
12. Update the output name parameters to descriptive and easy to find file names

```
library( terra )
library( sf )
library( doSNOW )
library( cppRouting )
library( RcppParallel )

## ----------------------------- USER INPUT --------------------------------

## Inputs
rdName                  <- "roads.shp"
rdField                 <- "MAX_MPH"
facilityName            <- "processingsites.shp"
facilityField           <- "PSITE_ID"
resourceName            <- "plots.shp"
resourceField           <- "PLOT_CN"
useCompiledCode         <- FALSE ## logical
rasterResolution        <- 50 ## meters
nIterations             <- 10
connectedNodeThreshold <- 0.75

## Outputs
roadRasterName          <- "rd_speed.tif"
movedFacilityName       <- "moved_mills.gpkg"
movedResourceName       <- "moved_plots.gpkg"
outCSVName              <- "cost.csv"


## --------------------------- END OF USER INPUT -----------------------------

```

Figure #13 – User input section of the transportation cost R script with sample inputs

13. Save the transportation cost R script
14. If you are not in a new session, execute **`rm(list=ls()); gc()`** at the R or RStudio console/command line to remove any object present in the session. This is important for systems with limited RAM.
15. Execute **`source("transportation_cost.R")`** at the R or RStudio console/command line

The amount of time it takes for the script to run is dependent on the size of the datasets and the specified raster resolution, but it will likely take several minutes.

Fixing disconnected and orphaned road segments

Each execution of the transportation cost R script generates two geopackages containing disconnected (dc_file) and orphaned (orphan_file) road segments. It may be helpful to rename these files to keep track of versions. These geopackages contain points that follow disconnected and orphaned road segments. Disconnected and orphaned road segments result from omissions and discrepancies in the source road layers, and depending on available road data sources, it is not uncommon for there to be many disconnected and orphaned segments, especially in areas far from the urban street grid. Disconnected and orphaned road segments that are near plots or processing sites have the potential to introduce error and should be resolved before completing the workflow. Follow these steps to resolve both kinds of road segments:

1. Add the disconnected road segments and orphaned road segments geopackages to the ArcGIS project that contains your plot, processing site, and roads data
2. In the Contents pane, remove all layers except for the final plot, processing site, and roads layers as well as the disconnected and orphaned road segments layers you just added
3. Changing the basemap to an imagery basemap can aid with investigating the cause of disconnection anomalies
    a. On the Map tab in the Layer group, click <Basemap>
    b. Select your desired basemap. Imagery and Imagery Hybrid work well
4. On the Map tab in the Selection group, click <Select by Location> ()
5. Set the Input Features parameter to the disconnected road segments layer
6. Set the Selecting Features to the plots layer
7. Set the Relationship parameter to “Within a distance”
8. Set the Search Distance parameter to 500 meters or another value, if needed
9. Ensure the Selection Type is set to “New selection”
10. Click <OK> to run the tool

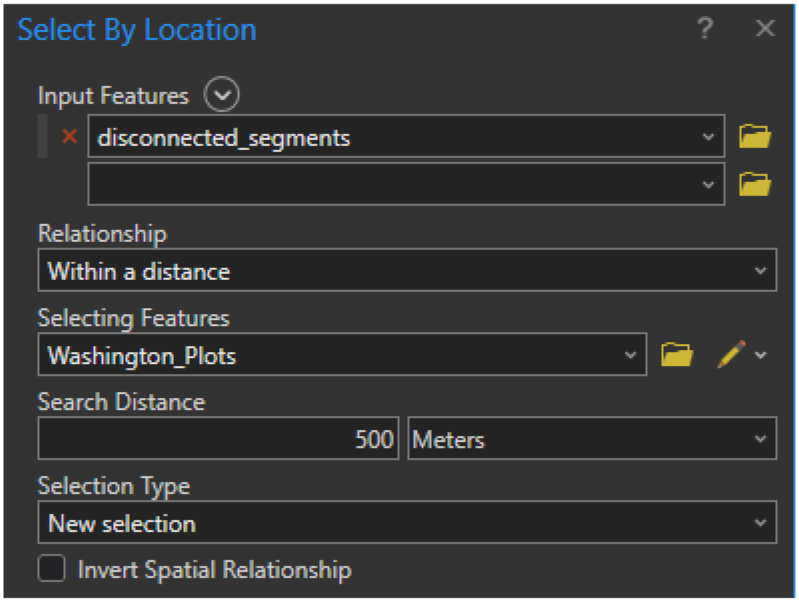


Figure #14 – ArcGIS Select by Location tool for a new selection

11. Open the Select by Location tool again
12. Keep the Input Features parameter as the disconnected road segments
13. Set the Selecting Features to the processing sites layer
14. Keep the Search Distance parameter the same
15. Ensure the Selection Type is set to "Add to the current selection"
16. Click <OK> to run the tool

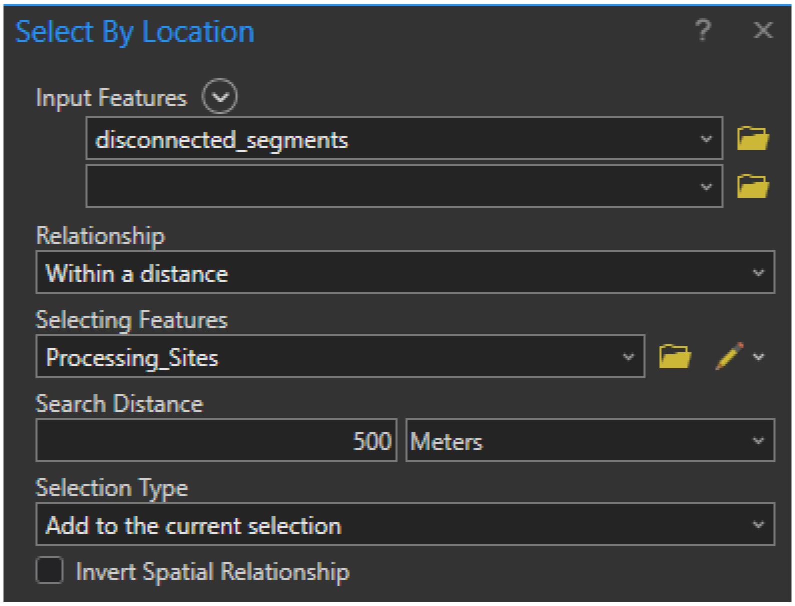


Figure #15 – ArcGIS Select by Location tool for adding to current selection

17. With the desired disconnected road segment points selected, open the Export Features tool in the Conversion toolbox
18. Set disconnected road segments as the Input Feature and confirm that the tool generates notification that the input has a selection
19. Provide a name for the tool output, for example, "selected_disconnected_segments"
20. Click <OK> to run the tool
21. In the Contents pane, Right click on the name of the selected disconnected segments layer just created and click <Open Attributes Table>
22. Double click the first line in the table
23. If needed, right click the line and select <Zoom to Selection>
24. If it is obvious, for example from consulting the base map image, that a road segment is missing from the road feature class, you will need to add it to the road feature class by digitizing. On the Edit tab in the Features group, click <Create> ()
25. In the Create Features pane, click the road layer to open the editing toolbar

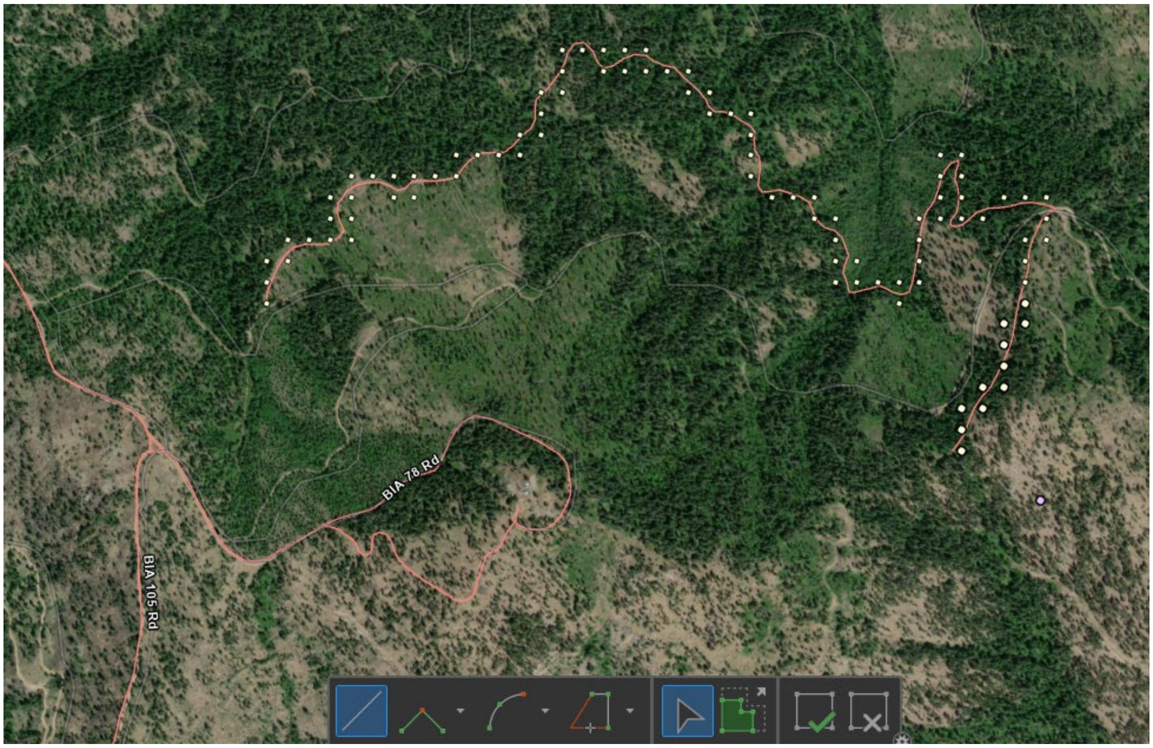
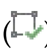


Figure #16 – Example of disconnected road segment, represented in the geodata package as both a line and the points used for the graph that represents that segment, that requires correction

26. Hover the cursor over the end of the disconnected segment until it snaps to the endpoint
27. Click to add a vertex
28. Continue digitizing the missing road segment following the basemap evidence of its existence ending at the road that is connected to the rest of the full road network

Figure #17 – Example of newly created line segment to fix disconnected segment

29. Click <Finish> () in the editing toolbar
30. In the Contents pane, select <List by Selection> ()
31. Unselect every layer except the selected disconnected segments layer

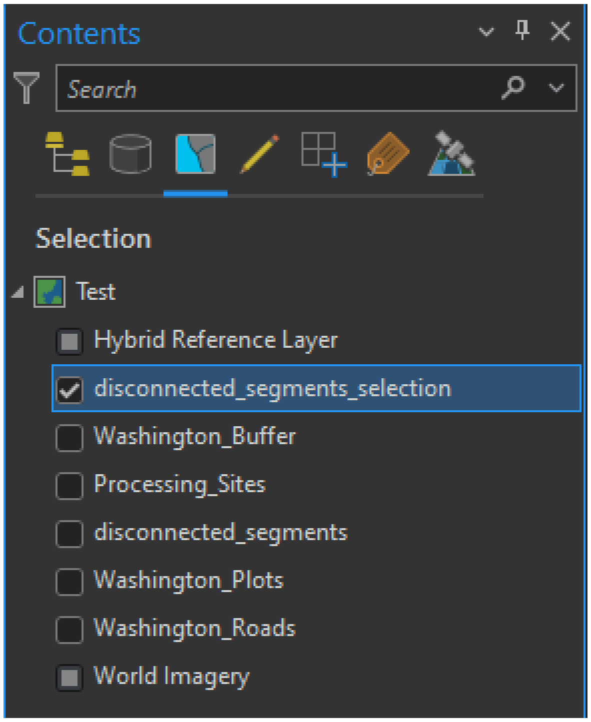


Figure #18 – ArcGIS Contents pane with only the selected disconnected segments layer selectable

32. On the Map tab in the Selection group, click <Select> <Lasso>
33. Lasso the selected disconnected segments points associated with the road that you have just resolved by establishing a line connecting the formerly disconnected segment to the rest of the road network
34. Use the delete button on the keyboard or click <Delete> in the Features group of the Edit tab
35. Note that some disconnected segments don't affect travel times because they represent roads that provide no advantageous route from plot to processing site, as shown below. In these cases, no road segments need to be added and the disconnected segments points can just be deleted

Figure #19 – Example of disconnected road segment that does not need to be corrected. As the road segments shown starting with the highlighted, cyan dot are farther from the plot (pink dot) than Emerald Lake Way, they will not impact travel times and can be deleted

36. Repeat steps 22 - 35 until all points from the selected disconnected segments layer have been analyzed and deleted (and new road segments digitized where needed)
37. Repeat steps 4 - 36 for the orphaned road segments
38. Click <Save> in the Manage Edits group on the Edit tab
39. For any newly created road segments, a MAX_MPH value must be assigned. On the Map tab in the Selection group, click <Select by Attributes> ( )
40. Set the Input Features parameter to the roads layer
41. Create a where clause to select road segments where MAX_MPH is null
42. Ensure the Selection Type parameter is set to “New selection”
43. Click <OK> to run the tool

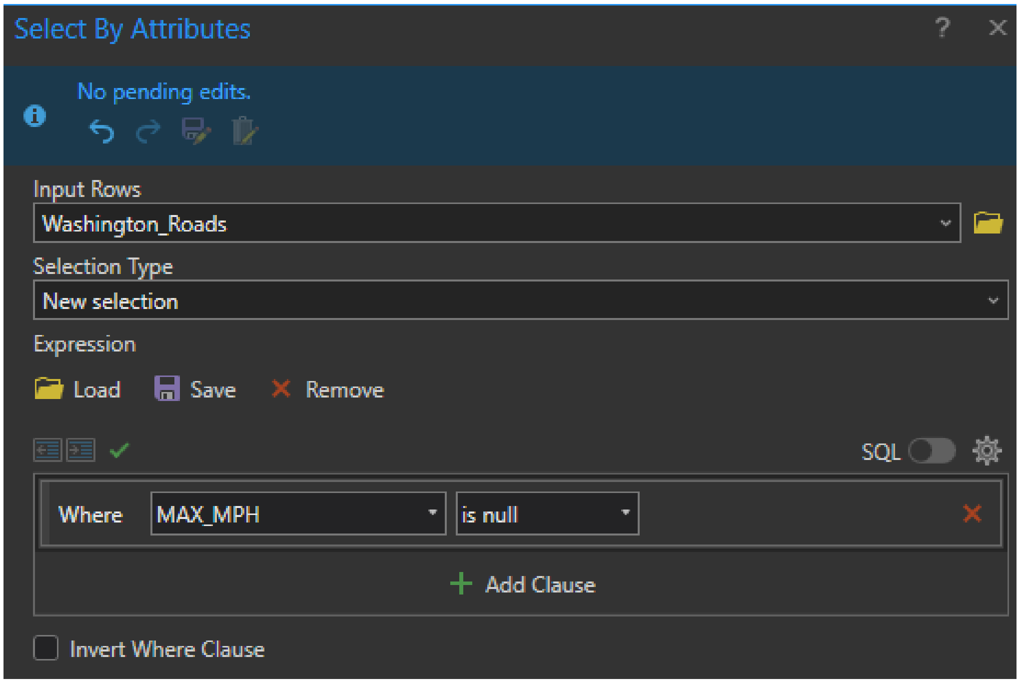


Figure #20 – ArcGIS Select by Attributes tool

44. With the new road segments selected, right click the roads layer in the Contents pane and select <Open Attributes Table>
45. Click <Show Selected Records> () at the bottom left of the table view
46. Right click the first line and select <Zoom to Selected>
47. Analyze the information available in the basemap at the location of the new road segment to estimate a plausible MAX_MPH value. Many of these will be narrow, unsurfaced roads where MAX_MPH is likely no more than 10 or 15
48. Double click the MAX_MPH cell and type the selected value, double checking as you go that the intended value was successfully entered
49. Continue following steps 46 - 48 as you move down the table until all lines have a MAX_MPH value assigned
50. Click <Save> in the Manage Edits group on the Edit tab
51. Open the Feature Class to Shapefile tool in the Conversion toolbox
52. Set the input to roads layer
53. Click <OK> to run the tool

Rerunning the transportation cost R script (after road corrections)

After the road data has been fixed and exported as a shapefile, rerun the travel times R script. The script will ask you if you want to overwrite any existing output files. If you do not want to overwrite the output files, first change their names in the user inputs. Again, ensure the work directory is set to the desired folder and that any other software that might have opened lock files to support their access to the input files has been shut down, or at least closed out all files and their associated file locks.

- Save the transportation cost R script
- If you are not in a new session, execute **`rm(list=ls()); gc()`** at the R or RStudio console/command line to remove any object present in the session
- Execute **`source("transportation_cost.R")`** at the R or RStudio console/command line

Adding data to the database

The database at the end of this workflow will contain data from the plots, processing facilities, travel cost data derived from the CSV file produced by the travel times R script and moved plots geopackage produced by the travel times R script. The travel times database creator R script processes the specified inputs and to create and populate these tables.

1. Ensure the plots and processing facilities CSV files have the information outlined in the "Preparing plot and processing site data" section above
2. Open your ArcGIS project that contains all your data
3. In the Catalog pane, navigate to the moved plots geopackage created by the transportation cost R script
4. Add the moved plots geopackage to the project
5. Right click on the moved plots layer in the Contents pane and select <Open Attributes Table>
6. Select <Export>() in the table view and save the attribute table as a CSV

7. Update the `costcsv` parameter to the name of the cost CSV created from the transportation_cost script

8. Update the `plotcsv` parameter to the name of the plot CSV used to create the plot point feature class

9. Update the `psitecsv` parameter to the name of the processing sites CSV used to create the processing site point feature class

10. Update the `movedistcsv` parameter to the name of the moved plots table exported in step 6 of this section

11. If you do not want to delete travel time records with times above a maximum travel time value, set the `pareflag` parameter to FALSE

12. If the `pareflag` parameter is TRUE, use the `maxtime` parameter to set the maximum travel time value for records to retain in the database

13. If you have run the script before, either delete the previously created database or rename it

```
library(DBI)
library(RSQLite)
library(tidyverse)

## --------------------- User Input ---------------------

# Cost CSV file you want to add to the database
costcsv <- "cost.csv"

# Plot CSV file you want to add to the database
plotcsv <- "plots.csv"

# Processing site CSV file you want to add to the database
psitecsv <- "processingsites.csv"

# CSV file with move distances from moved_plots geopackage
movedistcsv <- "moved_plots.csv"

# set to TRUE to get rid of records over a certain travel time
# set to FALSE to keep all records
pareflag <- TRUE

# max travel time used to pare down table
maxtime <- 5

## --------------------- End User Input ---------------------

```

Figure #21 – User Input section of database creation script with sample inputs

14. Save the travel times database creator script
15. Run the script or execute **`source("travel_times_database_creator.R")`** in the R or RStudio console/command line
16. Once the script has fully exectuted, examine and QA the gis_travel_times_master.db that it created to ensure it contains the expected information