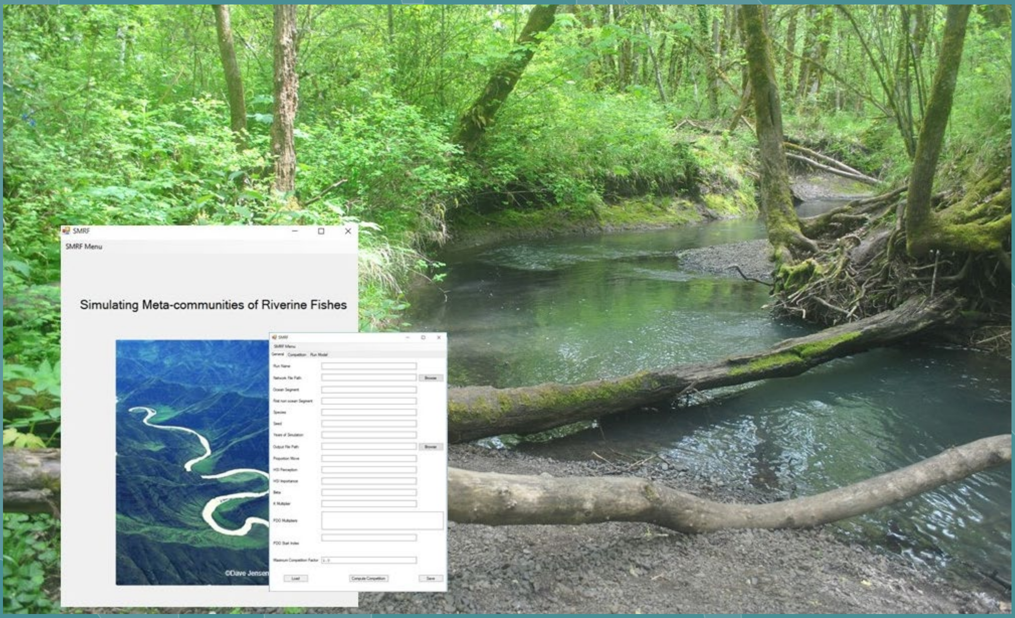
United States
Environmental Protection
Agency

EPA/645/R-20/001 | October 2019 | www.epa.gov/research

# Simulating Metacommunities of Riverine Fishes (SMRF) User Manual



Office of Research and Development
Center for Public Health and Environmental Assessment / Pacific Ecological Systems Division

EPA/645/R-20/001
October 2019

# Simulating Metacommunities of Riverine Fishes (SMRF) User Manual

by

Brenda Rashleigh[1], Allen Brookes[2], George Boxall[3], Joseph Ebersole[2],
Marcia Snyder[2], Joan Baker[4], Dennis White[4], Brandon Waller[2]

[1]US Environmental Protection Agency – Office of Research and
Development, Center for Public Health and Environmental Assessment
Narragansett, RI 02882
[2]US Environmental Protection Agency – Office of Research and
Development, Center for Public Health and Environmental Assessment
Corvallis, OR 97333
[3]Amnis Opes Institute
Bend, OR 97701
[4]US Environmental Protection Agency – retired

Project Officer
Allen Brookes

Pacific Ecological Systems Divison
Center for Public Health and Environmental Assessment
Corvallis, OR 97333

## Disclaimer Statement

This document has been reviewed in accordance with U.S. Environmental Protection Agency, Office of Research and Development, and approved for publication.   The views expressed in this manual are those of the author(s) and do not necessarily represent the views or policies of the U.S. Environmental Protection Agency. Any mention of trade names, products, or services does not imply an endorsement by the U.S. Government or the U.S. Environmental Protection Agency. The EPA does not endorse any commercial products, services, or enterprises.

This document provides links to non-EPA web sites that provide additional information about this topic. EPA cannot attest to the accuracy of information on that non-EPA page. Providing links to a non-EPA Web site is not an endorsement of the other site or the information it contains by EPA or any of its employees. Also, be aware that the privacy protection provided on the EPA.gov domain (see Privacy and Security Notice) may not be available at the external link.

# Abstract

Fish communities in river networks provide significant ecosystem services that will likely decline under future land use, human water demand, and climate variability. Modeling can be used to assess the consequences to multiple populations of one or more fish species from multiple stressors across a river network. We propose a modeling approach that is of intermediate scale and complexity. The model is spatially-explicit and age-structured, with three components: habitat suitability; population dynamics, including species interactions; and movement across a spatial network. Although this model is simple, it can form the basis of fisheries assessments and may be incorporated into an integrated modeling system for watershed management and prediction. The approach provides a heuristic tool for identifying critical data gaps in our understanding of watershed-scale fish-habitat relationships, particularly as these may be influenced by species behaviors and interactions. Model results provide testable hypotheses regarding species distributions and projected fish population responses to environmental change, water consumption, species invasions, and land use effects on water temperature.

# Foreword

The U.S. Environmental Protection Agency (EPA) is charged by Congress with protecting the Nation's land, air, and water resources. Under a mandate of national environmental laws, the Agency strives to formulate and implement actions leading to a compatible balance between human activities and the ability of natural systems to support and nurture life. To meet this mandate, EPA's research program is providing data and technical support for solving environmental problems today and building a science knowledge base necessary to manage our ecological resources wisely, understand how pollutants affect our health, and prevent or reduce environmental risks in the future.

The Center for Public Health and Environmental Assessment (CPHEA) provides the science needed to understand the complex interrelationship between people and nature in support of assessments and policy to protect human health and ecological integrity.

Fish communities in river networks provide significant ecosystem services that can be responsive to water quality and watershed conditions. We developed a model that simulates the consequences to multiple populations of one or more fish species – a metacommunity – from multiple stressors across a river network. The model is spatially-explicit and age-structured, with three components: habitat suitability; population dynamics, including species interactions; and movement across a spatial network. Although this model is simple, it can form the basis of fisheries assessments and may be incorporated into an integrated modeling system for watershed management and prediction.

## Table of Contents

# I) BACKGROUND

SMRF (Simulation of Metacommunities for Riverine Fishes) is a computer model for stream networks that models the population dynamics of interconnected fish communities through space and time. SMRF is mechanistic, population based, age-structured, and spatially-explicit. Model implementation consists of three primary parts: a program to create a stream network distance matrix, ComputeDistance.exe, a program to create Habitat Suitability Index (HSI) input files, ComputeHsi.exe, and the model runner, modelThree.exe. Each model component, as well as additional components, is described in greater detail later in this manual.

The basis for SMRF is a user provided virtual network comprised of individual stream reaches with associated physical attributes, which collectively represent a real-world system of interest. Networks can be exclusively freshwater or include ocean connectivity for migratory species. Potential habitat use is determined by life history characteristics and HSIs that limit fish distribution based on species-specific preferences and tolerance for key environmental variables. Realized habitat use (occupancy) is a combination of network suitability, inter/intraspecific competitive interactions, and the extent to which a species is able to move throughout the system. By default, SMRF uses mean annual flow (an indicator of stream size), seasonal temperatures, and the longitudinal gradient of stream reaches (slope) to determine potential habitat use. More advanced users can choose to accept these defaults or add/remove predictor variables to reflect the physical processes that govern fish distribution in their system of interest and for their chosen assemblage. Default predictors were developed to reflect riverine systems along the west coast of North America.

SMRF estimates the abundance of each species/age-class in individual stream reaches that are delineated by unique ID numbers. Estimates are produced seasonally at each ‘time step’ in the model. By default, there are three time steps each year (1 = fall/winter, 2 = spring, 3 = summer), but users can manipulate this sequencing to reflect seasonality more relevant to their chosen assemblage or local climatic conditions. Raw model output is reported as absolute abundance for each species/age-class in each stream segment, though interpretation of results should focus more on relative spatial distribution and population trends over time. Users can summarize output and produce abundance and density estimates for individual age-classes, multiple age-classes, or entire populations. Users may also specify whether to report results at a discrete time step or some combination of steps, in a specified stream reach or for the entire system.

The SMRF distribution package includes a suite of executable (.exe) and input (.xml) files in the form of a zipped working directory. A sample stream network (Calapooia River, OR) and fully parameterized input files are included which allow users to run basic simulations with minimal alteration. Additional species may be added to the modeled assemblage by performing research to compile relevant and justifiable life-history characteristics and environmental sensitivities for the new species. Default SMRF species have been parameterized from a combination of values derived from peer-reviewed journal articles, government reports, and logistic regression analyses performed on fish sampling data for species/variables where sufficient evidence was not available in the published literature.

Users can implement SMRF through either the included Graphic User Interface (GUI), or by running model components directly from the command line. Advanced users seeking to model large numbers of scenarios for comparison or sensitivity analyses will likely be better served using command line automation. SMRF uses many different files to run the model and generate output. It is important that these files are properly compiled and saved with the correct file extension for the model to run successfully and generate valid output. More detailed descriptions of the modeling directory file structure, the GUI, and each component file is outlined in subsequent sections of this manual.

SYSTEM REQUIREMENTS:

- Computer running a Microsoft Windows-based operating system
- Additional software needs:
    - Microsoft C++ Runtime Library (*available for download at https://support.microsoft.com/en-us/kb/2977003#bookmark-vs2015*)
    - Microsoft .NET Framework (*available for download at https://dotnet.microsoft.com/download/dotnet-framework/net48)*
    - Archive extraction software installed on your computer (e.g. WinZip, 7-Zip)
- SMRFv1.0_DistributionPackage.zip

# II) THE SMRF WORKING DIRECTORY

The SMRF model is distributed in the form of a zipped working directory containing the core executable programs and template input files required to create a customized model scenario. What follows is a description of the contents and structure of the default SMRF working directory. In many cases SMRF requires specific files to be read from designated locations and users should exercise caution when organizing this directory and avoid needlessly renaming core files or folders.

*Note: model-generated files are covered in greater detail in sections X and XI of this manual.*

SMRF – main directory where SMRF model files are unzipped includes bin, documentation, network, species, runs, and output directories. If this folder is named anything other than SMRF, or if it is nested inside another folder named SMRF, the model may not run properly.

- bin – core model components
  - SMRFGui.exe – Graphic User Interface (GUI) allowing users to manipulate input files and run SMRF in a user-friendly visual environment
  - ComputeDistance.exe – generates spatial geometry files based on the source network data (user provided)
  - ComputeHsi.exe – generates species-specific Habitat Suitability Indices (HSI) for segments in the virtual stream network
  - modelThree.exe – core model file that runs the SMRF scenario
  - modelLib.dll – required peripheral model file for modelThree.exe
  - TransformOutput.exe – summarizes raw model output into a .csv file
  - NetworkTools.exe – creates a copy of a network with all large reaches separated into smaller reaches

- Documentation – support documents and other model resources
  - SMRF_User.Manual.pdf
  - SMRF_Example.Run.pdf – Quick visual guide for running the SMRF Model
  - SMRF_Quickstart.Guide.pdf – Beginner's guide to all of SMRF's primary functions
  - Taxa Traits Database – Table classifying fish species by ecological traits (habitat and trophic guilds) widely used to compute generalized competition parameters
  - MapScript.R – R script that reorganizes a transformed output from a split network to a whole network
  - Output_Figures.R – R script that produces basic figures describing the output of a SMRF run

- network – repository for stream network files

  *Each network subfolder (if multiple) should include the files detailed below. A functional example network (Calapooia River, OR) has been included in the default directory. A template subfolder,* SMRF/network*/MyNetworkName, has been provided as a location for users to prepare their own stream network for custom SMRF simulations. Users can rename this folder to reflect their system of interest.*

  Network subfolders for stream network source data should include:
  - Network file(s) – in .dbf or .csv format (user provided, see section IV)
  - distance.xml – input file for the Network Generator
    - Stored in MyNetworkName folder
    - Used in conjunction with ComputeDistance.exe

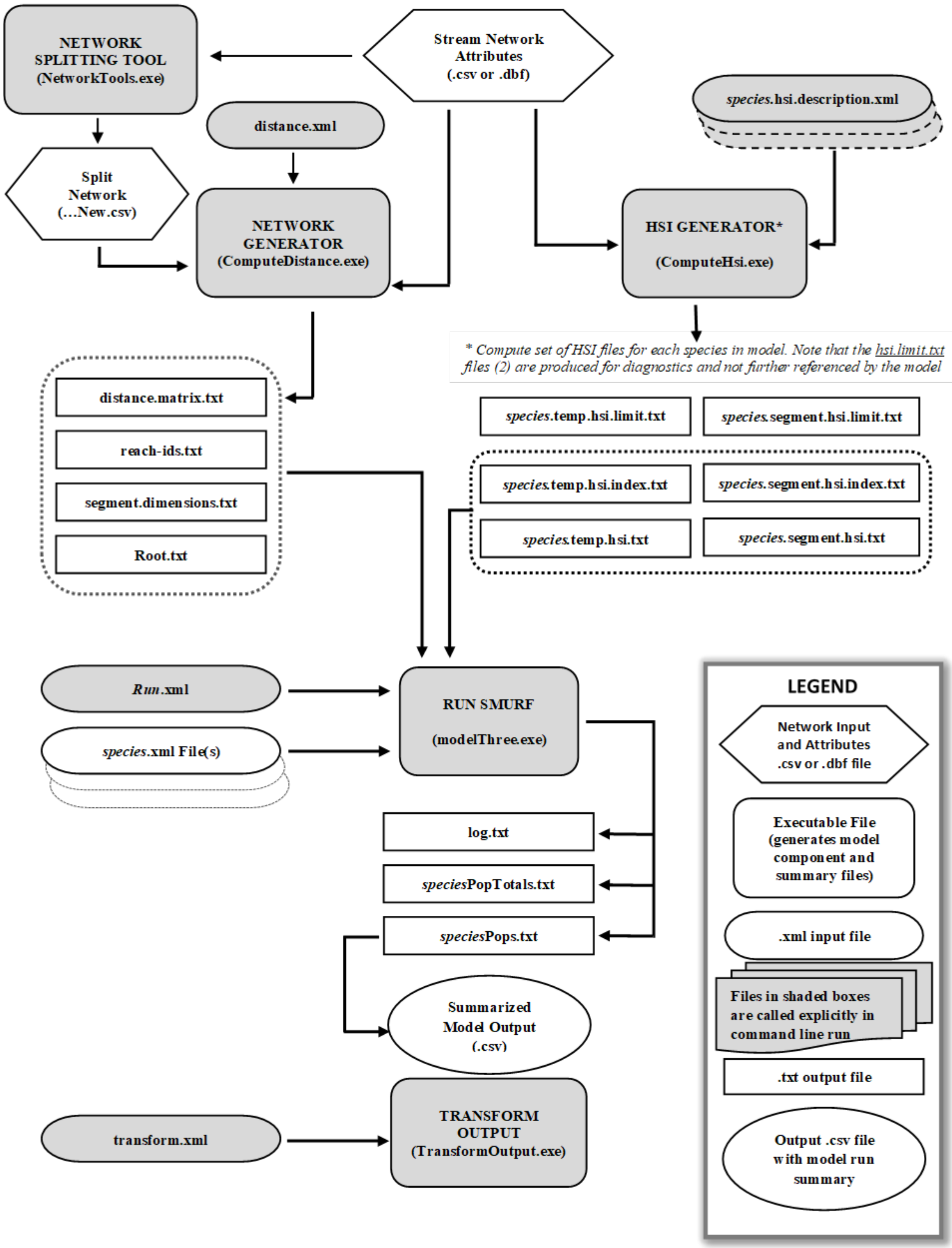
    - *Output from the SMRF Network Generator. See section X of this manual for explanation of model generated files.*

- species – species-specific input parameters and model generated network suitability
    - *species*.xml (multiple) – contains species-specific life history parameters (e.g. fecundity, survival, migration). Individual files are needed for each species included in model run (i.e. chinook.xml, cutthroat.xml, pikeminnow.xml, etc.)
    - *species*.hsi.description.xml (multiple) – defines species-specific habitat suitability (HSI) curves by season, stream segment attributes, and age-class. Determines potential habitat use. Individual files are needed for each species included in model.
        - Used in conjunction with ComputeHsi.exe
    - Species Library (i.e. SMRF/species/Species.Library) – storage to preserve master copies of species-specific *species*.xml model input files (listed above).
    - *Model output from the HSI Generator will also be written to the main* species *folder. See section X of this manual for explanation of model generated files.*

- runs – model initialization files to run different scenarios
    - Run.xml – input file that specifies run duration, species to include in the fish assemblage, and other initial parameters
        - Used in conjunction with modelThree.exe

- output – where raw model output and transformed output is stored
    - transform.xml – input file with instructions for summarizing raw model output data into a .csv file
        - Used in conjunction with TransformOutput.exe
    - Model output subfolders (model generated). *See section XI of this manual for explanation of model output files.*

# III) MODEL MECHANICS


NETWORK SPLITTING TOOL (NetworkTools.exe)
Stream Network Attributes (.csv or .dbf)
species.hsi.description.xml
distance.xml
Split Network (…New.csv)
NETWORK GENERATOR (ComputeDistance.exe)
HSI GENERATOR* (ComputeHsi.exe)
distance.matrix.txt
reach-ids.txt
segment.dimensions.txt
Root.txt
species.temp.hsi.limit.txt
species.segment.hsi.limit.txt
species.temp.hsi.index.txt
species.segment.hsi.index.txt
species.temp.hsi.txt
species.segment.hsi.txt
Run.xml
species.xml File(s)
RUN SMURF (modelThree.exe)
log.txt
speciesPopTotals.txt
speciesPops.txt
Summarized Model Output (.csv)
transform.xml
TRANSFORM OUTPUT (TransformOutput.exe)
LEGEND
Network Input and Attributes .csv or .dbf file
Executable File (generates model component and summary files)
.xml input file
Files in shaded boxes are called explicitly in command line run
.txt output file
Output .csv file with model run summary

** Compute set of HSI files for each species in model. Note that the hsi.limit.txt files (2) are produced for diagnostics and not further referenced by the model*

## The Network Generator – `ComputeDistance.exe`

Movement in SMRF is based on distances between stream reaches. This is accomplished in the model using a matrix indicating the distance from each individual reach to every other reach in the network. `ComputeDistance.exe` creates this matrix, `distance.matrix.txt`, using a user-provided input file containing columns describing network connectivity (from node and to node), reach length, cumulative drainage area, and a unique reach ID, along with the physical habitat variables that will be referenced later by other parts of the model. Much of these data can be obtained directly from the National Hydrography Dataset (NHD, NHDPlus) databases, but any .dbf or .csv file that contains this information will work. See section V of this manual for more specific network source data requirements and additional resources.

`ComputeDistance.exe` also creates a file, `segment.dimensions.txt`, containing the width, length, and area of each network reach. These dimensions are used by the model for computing carrying capacity. A third file produced, `reach-ids.txt`, is an association of reach index to the ID given in network input file. `Root.txt`, an index file indicating the root of the network tree, is also generated during this process.

`ComputeDistance.exe` takes a single argument, `distance.xml`, a file that describes the source file used, and the output files to be created.

In some networks, the distance between two stream reaches will be too great for a species to cross. To combat this, `NetworkTools.exe` is used to divide all lengthy reaches into easily spannable segments. See Section IX for more details on the Network Splitting Tool.

## The HSI Generator – `ComputeHsi.exe`

The Habitat Suitability Index (HSI) is a number between 0 and 1 indicating the relative quality of a reach for an individual of a particular species at a particular life stage. SMRF uses reference tables to supply these indices. The tables are contained in a set of HSI files that need to be generated for each modeled species. `ComputeHsi.exe` is used to create files containing these HSI tables and requires input from *`species`*`.hsi.description.xml`, a file that describes each of the HSI functions needed for a single species.

The xml file contains a collection of all the individual piece-wise linear *functions* assigning HSIs for each predictor variable that could apply to a species throughout their life history (constituent functions). These functions are followed by multiple HSI *function sets*, each containing one or more HSI functions that apply to a particular age-class/season. There are both function sets that specify the overall reach suitability based on all predictors combined (*`species`*`.segment.hsi`), and function sets that determine suitability for temperature alone (*`species`*`.temp.hsi`).

HSI function sets consist of a combining function, a set of constituent function references, and a set of use descriptions. At this time, the only combining function type is “min” (minimum). For the type “min,” the value of the function for a given input is the minimum of the values of the constituent functions. An HSI function set must include unique use descriptions for each age/season in the lifecycle to which it applies and can apply to multiple ages and season.

See section VI of this manual for more information on the contents of the *`species`*`.hsi.description.xml` file and proper syntax for coded functions, and section X for a description of output produced by the Network Generator.

### Model Run – `modelThree.exe`

The core model program is called `modelThree.exe` and takes a single input; an xml file that describes the stream network and species to which the model run applies (`Run.xml`). The model presumes a structured workspace when searching for network and species files, which includes a runs directory containing any number of model input files, a network directory with any number of subfolders (one for each stream network) containing source data and model generated network files, and the `species` directory containing sets of model generated HSI files and species description files for each modeled species. Section II of this manual describes the structure and contents of the SMRF working directory in more detail.

See section VI of this manual for additional information on the contents of the *`Run`*`.xml` file, and section XI for a description of output produced by running the SMRF model.

### Transform Model Output – `TransformOutput.exe`

The result of running SMRF is a group of files (`.txt` format) containing detailed information on the dynamics of the modeled fish populations over time. Users can 'transform' this data into a format that can be more easily manipulated for analysis and mapping (`.csv`). This is accomplished using the program `TransformOutput.exe` which takes as input a single file: `transform.xml.` This file describes the location of the network files, identifies the raw model output to be transformed, names the `output.csv` file, and specifies which age-classes to summarize and for what time step(s) in the model.

See section VI of this manual for additional information on the contents of the *`transform`*`.xml` file and data transformation options. Section XI gives a description of the raw output produced by running SMRF, as well as the final output produced by `TransformOuput.exe`.

## IV) STREAM NETWORK SOURCE DATA REQUIREMENTS/FORMAT (*User Provided)*

To initialize and run a custom SMRF model scenario, users must provide a file or set of files describing the spatial attributes and physical characteristics of their system of interest. At minimum these data should include the geometric dimensions of individual stream reaches, network connectivity, and information on key environmental variables (mean annual flow, slope, and seasonal temperatures).

SMRF is compatible with source network data stored in either a `.csv` file or the `.dbf` component of an ESRI shapefile (GIS). If you choose to work with a `.dbf` network file, it is good practice to copy all other associated shapefile components to the network subfolder (e.g. *`Your_Network`*`.dbf` *plus* *`Your_Network`*`.shp`, *`Your_Network`*`.shx`, *`Your_Network`*`.prj`, etc.). This will facilitate spatial analyses of model output in a GIS (e.g. Q-GIS, ArcGIS).

The most accessible means of creating a network for simulations applied in the continuous United States is by clipping the area of interest from the National Hydrography Dataset (NHDPlusV2). This database provides the spatial geometry for stream lines labeled with a unique COMID number, as well as corresponding attribute values for mean annual flow and stream gradient (slope). NHDPlusV2 can be accessed online at:

http://www.horizon-systems.com/nhdplus/NHDPlusV2_home.php (external link)

For the SMRF network to function properly, the associated shapefile needs to be a single linear vector free from side channels, braids, and any other hydrological abnormalities. In some portions of the NHD dataset, the linear vector of the stream network will include these issues. If this is the case, users can consult the National Stream Internet Project (provided by the Rocky Mountain Research Station) to download a flowline that is cleaned and free of abnormalities. Otherwise, the user can manually edit their NHD data to fix the issues themselves. The NSI data can be accessed online at:

https://www.fs.fed.us/rm/boise/AWAE/projects/NationalStreamInternet/NSI_network.html

Users will need to provide seasonal temperature estimates for the stream segments in their network if they choose to use temperature as a network attribute. Spatial Stream Network (SSN) models can be a useful tool for estimating temperature or other stream networks across an entire network based on observed statistical relationships.

https://www.fs.fed.us/rm/boise/AWAE/projects/SpatialStreamNetworks.shtml

Established models may be available for some portions of the US; see for example:

https://www.fs.fed.us/rm/boise/AWAE/projects/NorWeST.html

In some cases, a network may be assembled where a single stream reach is longer than a species' move distance value. This creates a growing mass of that species, since they can enter this reach and are not able to leave. To combat this, the size of all reaches that are too long must be divided into small, traversable reaches using the Network Splitting Tool. See section X for how to use the Network Splitting Tool.

Specific Source Data Requirements*

** Field names noted below (underlined) are the column labels found in the provided Calapooia network used in the SMRF Example Run (SMRF/network/Calapooia/CalapooiaModel.dbf). Users can adopt their own naming conventions for data fields, provided they are correctly specified in the `distance.xml` input file for the Network Generator, or the `species.hsi.description.xml` input for the HSI Generator. See section VII of this manual for more details.*

- Unique ID number identifying each stream reach (referenced in `distance.xml`)
  - COMID from NHDPlusV2
- Spatial connectivity identifiers (referenced in `distance.xml`)
  - TNODE_ – 'to node'. Point indicating one extreme (spatial) of the stream reach
  - FNODE_ – 'from node'. Point indicating the opposite extreme of the stream reach
  - *Note: networks created in a GIS (ESRI shapefile .dbf) should inherently contain topological data on nodes and connectivity for stream lines.*
- Network geometry (referenced in `distance.xml`)
  - LENGTH – reach length in meters
  - CUMDRAINAG – cumulative drainage area for the stream reach in square kilometers
- Physical attributes (default SMRF predictor variables, referenced in HSI function definitions in `species.hsi.description.xml`)
  - MAFLOWU – mean annual flow (cubic feet per second) as an overall indicator of stream size
  - SLOPE – physical gradient (longitudinal) of the stream reach

    - TEMPERATURE – mean seasonal temperature (for fall/winter, spring, and summer)

- User-added attributes – if desired, additional or alternative predictor attributes can be added to the stream network (e.g., water chemistry, channel substrate, etc.). To be used in SMRF, HSI functions for the new attributes will need to be created within the `*species*.hsi.description.xml files.`

## V) RUNNING SMRF MODELS

In the distribution package, you can find additional documents designed to assist with running the SMRF model (`SMRF/Documentation`). If you are new to SMRF or unfamiliar with it, see SMRF_ExampleRun.pdf for a start-to-finish demonstration of setting up and running a single-species or assemblage model with included example files. If you are more experienced with the model and would like to personalize the results, See `SMRF_QuickstartGuide.pdf` for step-by-step instructions on creating a customized SMRF model using your own network data and fish assemblage.

Refer to the resources in the next section for assistance in creating and modifying the required model input files (.xml).

The SMRF Model can be run in two different ways. The first is using the SMRF Graphic User Interface, or `SMRFGui.exe`. See section VII for instructions on how to operate the GUI. The second is using the Command Line, which is described in detail in section VIII.

## VI) WORKING WITH MODEL INPUT FILES

** Input files called by the Network Generator (`distance.xml`), HSI Generator (`species.hsi.description.xml`), during model runs (`species.xml, Run.xml`), and for output transformation (`Transform.xml`) must be saved with the appropriate .xml extension to run in SMRF. For basic applications these files can be created/edited/saved directly through the GUI. More advanced users may choose to create/edit/inspect the raw code for input .xml files in a text editor (e.g. MS Wordpad, Tinn-R). Annotated copies of example input.xml files can be found at `SMRF/Documentation/XMLdescrip`*

**`distance.xml`** – (`SMRF/network/*YourNetworkSubfolder*`)

*Called by ComputeDistance.exe in the Network Generator*

Distance input files indicate the name and location of the network source data, column labels for required SMRF parameters, and the names/output locations of model generated network geometry files.

For proper syntax, see `SMRF/Documentation/XMLdescrip` for an annotated version of a complete `distance.xml` file.

The following are descriptions of each code segment present in a `distance.xml` file:

- `<Network Data>` – name and location of user provided source data
    - `fromColumn` – name of data field containing 'from node' for stream reach connectivity
    - `toColumn` – name of data field containing 'to node' for stream reach connectivity
    - `lengthColumn` – name of data field containing stream reach length in meters
    - `drainageColumn` – name of data field containing cumulative drainage values (square kilometers)

  - `idColumn` – name of data field containing unique stream reach ID numbers

- `<HydraulicGeometryCoefficient>` - Optional function, manually replacing the coefficient (*a*) in the width-calculation function. This function estimates channel width in meters as a function of cumulative drainage area ($km^2$) following the form

  $w = a(DA)^b$

  where w = width in meters, *DA* is cumulative drainage area in $km^2$, and *a* and *b* are the fitting coefficient and exponent, respectively, taken from Table 3 in Bieger et al. 2015. If not present, the value for *a* defaults to 2.70, the value for USA streams in Bieger et al. 2015 Table 3.

- `<HydraulicGeometryExponent>` - Optional function, manually replacing the exponent (*b*) in the width-calculation function. This function estimates channel width in meters as a function of cumulative drainage area ($km^2$) following the form

  $w = a(DA)^b$

  where w = width in meters, *DA* is cumulative drainage area in $km^2$, and *a* and *b* are the fitting coefficient and exponent, respectively, taken from Table 3 in Bieger et al. 2015. If not present, the value for *b* defaults to 0.352, the value for USA streams in Bieger et al. 2015 Table 3.

- `<DistanceMatrixFile>` – name of the output file specifying the distances between each unique network reach. *Note: file name should remain `distance.matrix.txt`, do not alter.*

- `<SegmentDimensionsFile>` – name of the output file containing geometry for each stream reach (width, length, area). *Note: file name should remain `segment.dimensions.txt`, do not alter.*

- `<IDFile>` – name of the output file containing unique segment ID numbers. *Note: file name should remain `reach-ids.txt`, do not alter.*

- `<RootFile>` - name of the output file indicating the root of the network tree. *Note: file name should remain `Root.txt`, do not alter.*

### ***`species.xml`*** – species description file (`SMRF/species`)

A species description file describes relevant life history characteristics and provides other species-specific information needed to run SMRF. These input files also contain the parameters needed to generate inter/intraspecific competition and identify the names and locations of the appropriate HSI files referenced during model runs.

SMRF provides parameterized files for 21 fish species which are included with the distribution package; Chinook salmon (*Oncorhynchus tshawytscha*), cutthroat trout (*O. clarkii*), steelhead/ rainbow trout (*O. mykiss*), northern pikeminnow (*Ptychocheilus oregonensis*), speckled dace (*Rhinichthys osculus*), redside shiner (*Richardsonius balteatus*), reticulate sculpin (*Cottus perplexus*), largescale sucker (*Catostomus macrocheilus*), smallmouth bass (*Micropterus* dolomieu), bluegill (*Lepomis macrochirus*), threespine stickleback (*Gasterosteus aculeatus*), and mosquitofish (*Gambusia affinis*).

Users may develop their own assemblage by researching the life-history parameters detailed below for each additional species and compiling this information into similarly formatted xml files. To avoid errors in naming convention or file-formatting, use a previously established species file as a template for any new species.

The following are descriptions of each code segment present in a species xml file:

- `<Name>` – the name of the fish species. Used to label the model output files (i.e. if Name = cutthroat, SMRF will produce final output files named cutthroatPopTotals.txt and cutthroatPops.txt).
- `<MaxAge>` – maximum age for an individual of this species. No individual can be older than this age.
- `<AgeMaturity>` – the age at which this species becomes an adult and able to reproduce.
- `<SpawnSeason>` – the season in which this species spawns. By default, SMRF models three seasons (1=Fall/Winter, 2=Spring, 3=Summer).
- `<Anadromous>` – binary indicator of anadromy. 1 = anadromous, 0 = resident.
- `<InitFraction>` – the value that is multiplied with each reach's carrying capacity to generate initial fish populations in the model.
- `<Fecundity>` – maximum number of eggs produced in a single spawning season. One entry for each year of life expectancy (space separated vector). Zero values indicate age-classes that have not yet reached sexual maturity.
- `<OceanReturn>` – for anadromous species only. The estimated fraction of an age-class population that will return from the ocean in a given year (space separated vector).
- `<SegmentHsi>` – name of the file containing the HSI table (slope and flow) for network reaches. Named/produced by HSI Generator. *See section X of this manual for explanation of model generated files.*
- `<SegmentHsiIndex>` – name of the file that indicates HSI functions referenced by the model for each time step in a species' life-history. Named/produced by HSI Generator. *See section XI of this manual for explanation of model generated files.*
- `<TempHsi>` – name of the file containing the temperature HSI table. Named/produced by HSI Generator. *See section X of this manual for explanation of model generated files.*
- `<TempHsiIndex>` – name of the file that indicates HSI functions (temperature) referenced by the model for each time step in a species' life-history. Named/produced by HSI Generator. *See section X of this manual for explanation of model generated files.*
- `<Habitat>` – species habitat preference (Water Column, Benthic, Not Specified)
- `<Trophic>` – functional feeding guild (carnivore, invertivore, herbivore, omnivore, Not Specified)
- `<Velocity>` – species velocity preference (rheophil, pool, other, Not Specified)
- `<MigrationIndex>` – for anadromous species. A matrix showing season at which migrations take place, 1 for out migration, 2 for in migration, 0 for no migration. *[rows = age-class (years), columns = time steps (seasons)]*
- `<SurvivalBase>` – a matrix assigning survival fractions for each year/season of life expectancy *[rows = age-class (years), columns = time steps (seasons)].*
- `<SurvivalOceanBase>` – a matrix assigning ocean survival fractions for each year/season of life expectancy *[rows = age-class (years), columns = time steps (seasons)].*
    - Note: Ocean survival is assessed once annually and only during step 3 (summer) as currently referenced by SMRF. However, a complete matrix must be present in the code for the model to function. Best practice to assign 0's to non-referenced matrix values
- `<SurvivalStdev>` – a matrix assigning survival standard deviations for each year/season of life expectancy. Corresponds to the values in the 'SurvivalBase' matrix. *[rows = age-class (years), columns = time steps (seasons)]*

- `<OceanSurvivalStdev>` – a matrix assigning ocean survival standard deviations for each season of life expectancy. Corresponds to the values in the 'SurvivalOceanBase' matrix. *[rows = age-class (years), columns = time steps (seasons)]*
- `<MoveDistanceUp>` – a matrix assigning the maximum distance (meters) a species can move upstream for each time step in the model (season of life expectancy). *[rows = age-class (years), columns = time steps (seasons)]*
- `<MoveDistanceDown>` – A matrix assigning maximum move distance downstream (meters) for each year/season of life expectancy (model time steps). *[rows = age-class (years), columns = time steps (seasons)]*
- `<CarryingCapacity>` — describes functions for computing carrying capacity. Consists of individual elements that cover each season of life expectancy. Each element can calculate capacity based on length, area, or bounded area (`type`, in m or m$^2$). If values apply to multiple ages and seasons, they can be put into the same CarryingCapacity element. Each element can have one or more `Apply` sub-elements that define the life stage (age/season) to which they apply. When only age is given, the function applies to all seasons. If age and season are given then the function applies to that age and season only.
    - `type` – method for deriving carrying capacity. Can be calculated based on "length", "area", or "bounded area". For type= "area", the 'multiplier' is multiplied by the reach area to get the capacity. For type= "length", the multiplier is multiplied by the length to get capacity. Type= "bounded area" is used when carrying capacity is applied differently depending on channel width. Under this type there can be multiple multipliers, applied based upon established bounds (widths).
        - Bounded area works by setting up an interval where carrying capacities are true.
        - *Example*: <bound threshold="0" multiplier=".6"></bound><bound threshold="20" multiplier=".45"> In this case, carrying capacity is set to 0.6 fish per sq meter for streams less than 20m wide, and 0.45 fish per meter in streams >20m wide. The first interval is from 0-20m in width, and the 2nd interval is anything greater than 20m. The multipliers for each interval are after the bound threshold.
    - `multiplier` – value that is multiplied by length, area, or bounded area (type) to calculate carrying capacity
    - `<Apply>`
        - `age` – the age-class (in years) to which the element applies
        - `season` – the time-time step (season) to which this element applies. 1 = fall/winter, 2 = spring, 3 = summer.

## ***species*.hsi.description.xml** – HSI description file (`SMRF/species`)

*Called by ComputeHSI.exe in the HSI Generator*

An HSI description file defines species-specific habitat suitability curves (HSI) by season, stream segment attributes, and age-class in order to determine potential habitat use in the stream network. Individual description files are needed for each species included in model run.

For proper syntax, see `SMRF/Documentation/XMLdescrip` for an annotated version of a complete *`species`*`.hsi.description.xml` file.

The following are descriptions of each code segment present in the HSI description file:

- `<species>`
  - `name` – The name of the corresponding species file, minus the file extension (i.e. "chinook.xml" would be entered as "chinook" in the code
- `<variables>`
  - `file` – name of file containing the physical habitat variables (generally the source network data file)
- `<function>`
  - `name` – names the function
  - `variable` – indicates the variable to which this function applies
  - `type` – type of function ("necessary" by default)
  - `<point>` – assigns HSI scores over the variable range.
    - `x` – variable value,
    - `y` – HSI score (0.0 to 1.0).
    - *Note: SMRF will interpolate between the specified values to generate an HSI curve. Most default SMRF HSI curves specify four points representing the lower/upper limits of species tolerance and the lower/upper optimum values for the habitat variable. More points can be used to shape the curve if desired. HSI curves should be well supported by evidence from either primary literature or analysis of empirical sampling data to determine species distribution (i.e. logistic regression analysis). Specificity should only reflect your level of certainty.*
- `<hsi>`
  - `name` – names the HSI file that will be produced, minus file extension (no .xml)
  - `path` – specifies the output location for the HSI file
  - `<hsiFunction>`
    - `name` – the name of the function to call
    - `<combiningFunction>`
      - `type` – indicates how multiple hsi functions are handled by the model. The default is a minimizing function (type= "min") where the least suitable of the variable functions (flow, slope, temp) limits distribution.
    - `<function>`
      - `name` – the name of the function to apply for this species, age-class, and season
    - `<use>`
      - `age` – the age-class to which this function applies
      - `season` – the annual time step (season) to which this function applies
      - *Note: functions may apply to multiple age-class/seasons by inserting additional <use> parameters in the code*

**`Run.xml`** – model scenario file (`SMRF/run`)

*Called by ModelThree.exe (model run)*

For proper syntax, see `SMRF/Documentation/XMLdescrip` for an annotated version of a complete `Run.xml` file.

The model scenario file contains values for each of the parameters necessary to run the model. Following is a description of each parameter:

- `<name>` – **not currently referenced by SMRF during model runs**. This parameter is intended to facilitate differentiation between output files produced by alternative model scenarios. This could be in the form of appending the specified <name> to file names (i.e. log.*name*.txt) or embedded as a header within output files.
- `<network>` – the name of the `SMRF/network` subfolder where the stream network definition files are located (produced by the Network Generator).
- `<ocean>` – the row number (in the network file) of the stream segment that represents the ocean. Optional, only needed for anadromous species. *Best not to delete even if the fish are not anadromous, you may decide to use migratory species later.*
- `<firstNonOcean>` – row number of the stream segment that represents the first segment that is not ocean.
- `<species>` – space separated list of species to include in the model run. Use the same name as the `species.xml` file (i.e. `steelhead.xml` would be entered as `steelhead`). These names will also be used to label model output files.
- `<seed>` – seed used to initialize random number generator
    - Same seed = same result. Change seed for variation if you desire stochasticity.
- `<nyears>` – number of simulated years for this model run
- `<output>` – name/path of the folder where output files will be written
    - Be sure to use unique output folder names for each model scenario to avoid overwriting data from previous runs
    - *Note: the folder does not need to exist prior to running the model, SMRF will create a folder with the specified name when writing output.*
- `<ProportionMove>` – proportion of non-surviving population held back for additional phased move
    - Currently not used. Do not delete because it still needs to be there for the model to run.
- `<HSIperceptionerror>` – reflects the ability of modeled fishes to perceive the quality of the environment (suitability). Value sets range from which to draw from normal distribution with mean = 1.
    - 0.0 = no error or perfect perception; 1.0 = high perception error
    - Used in movement code for the fish to redistribute based on habitat quality
- `<HSIimportance>` – multiplier for the move equation, default value = 1.0
    - Used as an exponent on reach HSI score in the movement calculation to modify the relative role of habitat suitability when scoring reaches for movement.
    - Increasing the value "attracts" fish more strongly to reaches with higher HSI scores, all else being equal
- `<Beta>` – empirical based exponent used for determining movement rates of fish.
- `<KMultiplier>` – Not currently used. Default value = 1.0

- `<PdoMultipliers>` – a vector of multiplier values for correcting ocean survival values for anadromous species to include the influence of the Pacific Decadal Oscillation (PDO).
    - Empirically derived
    - Entries for each year in the PDO cycle
    - Sequence repeats after reaching the final specified value
- `<PDOstartIndex>` – index of which PDO multiplier in the specified sequence to apply for the first model year.
    - Default = 0. The model will begin with the first value in the PdoMultiplier vector
    - PDOstartIndex = 2 indicates that the model will begin by referencing the second entry in the PDO vector
    - Allows for models to begin during a specific point in the PDO cycle
    - *Note: it is best to avoid changing these values without proper justification*
- Competition: competition and predation are defined as a reduction in carrying capacity of one species by another species. The amount of reduction is based a weight, and a conversion fraction. Competition and predation are defined using the `CompetitionRecords`. A `CompetitionRecords` contain one or more `CompetitionRecord` elements. `CompetitionRecord` elements contains the following parameters for two species.
    - Spp1: first species name. (Prey or losing competitor)
    - Spp2: second species name. (Predator or winning competitor)
    - BeginAge1: beginning age at which this record applies for species 1.
    - BeginSeason1: beginning season at which this record applies for species 1.
    - EndAge1: last age for which this record applies for species 1.
    - EndSeason2: last season for which this record applies for species 1.
    - BeginAge2: beginning age at which this record applies for species 2.
    - BeginSeason2: beginning season at which this record applies for species 2.
    - EndAge2: last age for which this record applies for species 2.
    - EndSeason2: last season for which this record applies for species 2.
    - Weight: How much an effect spp2 has on spp1
    - Limit: Maximum proportion of carrying capacity that can be reduced via competition

  The Age/Season records are generated according to the life stage information in the *species.xml*. Each of the records represent a life stage of that species, and every species' life stage gets matched with every other life stage/species combination over the course of the CompetitonRecords. For each species, there will be a record that spans from:
    - Age 1, [SpawnSeason] to the season before [JuvenileAge], [JuvenileSeason]
    - [JuvenileAge], [JuvenileSeason] to the season before [AgeMaturity], [MaturitySeason]
    - [AgeMaturity], [MaturitySeason] to [MaxAge], Season 3

  For species that mature rapidly, meaning they advance through multiple life stages within a single season, entering the life stage information with this overlap can cause issues when computing competition. In the case of *Mosquitofish*, the species progresses from spawning to maturity all within Age 1, Season 3, and attempting to compute competition like this results in erroneous life stages. To remedy this, it is recommended that the life stage advancements get spread out over the following seasons, so they are no longer stacked.

**`transform.xml`** – model output transformation file `(SMRF/output)`

*Called by TransformOutput.exe*

Transform files indicate the file path for the SMRF working directory, the location of the network files used in the run, and the name the file containing the model data that will be transformed. It also allows users to specify what age-classes and time step(s) to report in the transformed output file. Individual transform files must be created and run for each species included in the model for which population summaries are desired.

Output from the transformation is produced at a user defined time step in the model or can be reported as the mean value from an aggregation of steps in order to minimize the stochastic influence associated with looking at a discrete point in time. Values are reported as the absolute abundance (# fish), linear density (# fish/m), and area density (# fish/m$^2$), for each age-class in a species' life history. Users can choose to produce additional output columns that group age-classes together in order to reflect abundance by stage of development (e.g. fry, smolt, parr) rather than absolute age. For example, users might choose to produce estimates for the total adult Chinook salmon return (multiple ages) to the stream network averaged over the last five spring seasons (multiple time steps).

For proper syntax, see `SMRF/Documentation/XMLdescrip` for an annotated version of a complete `transform.xml` file.

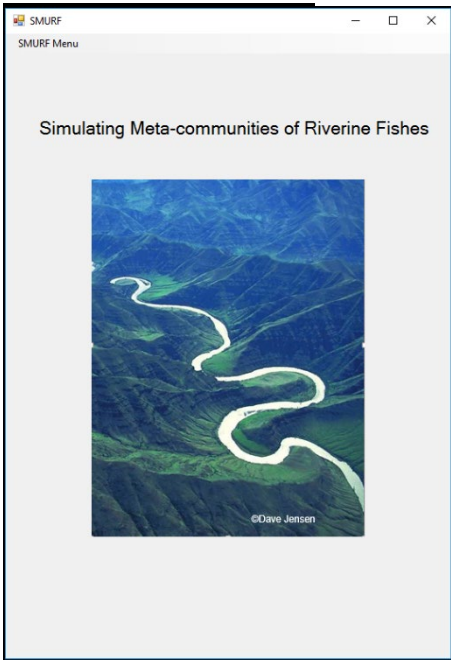
The following are descriptions of each code segment present in the model output transformation file:

- `<network>`
  - `runpath` - specifies the runpath for the working directory containing all the model files (i.e. `C:\SMRF`)
  - `networkPath` – indicates the subfolder containing the network files
- `<popCounts>`
  - `file` – the name of the `modelThree.exe` output file to be transformed (raw SMRF output in the form of *`species`*`Pops.txt`*)*
- `<output>`
  - `file` – names of the file to be created containing the transformed output with .csv extension (i.e. `cutthroatLast5springs.csv`)
  - `steps` – indicates time steps to report in the transformed output. Many methods exist to select timesteps for output
    - Continuous - steps 3-90 get selected if you use beginStep="3" endStep="90">
    - Discontinuous - only specific seasons; if you use steps="1 3 5" only steps 1, 3 and 5 are selected.
    - You can use the listed equations to find the season steps that you are interested in:
      - (#seasons/year)(year) – timestep to subtract to reach chosen season)
      - Winter steps=(3)(x)-2
      - Spring steps=(3)(x)-1
      - Summer steps= (3)(x)
    - Steps are **seasons, not years** and go from winter, spring, and summer.
- `<additionalOutput>`
  - `columns` – a vector for adding new output columns in the transformed data that summarize multiple combined age-classes. Absolute abundances, linear density, and area density are included. The number entered here correspond to the model age of the species being transformed and can be entered in the same formats used for `steps`.

## VII) THE SMRF GRAPHIC USER INTERFACE (GUI)

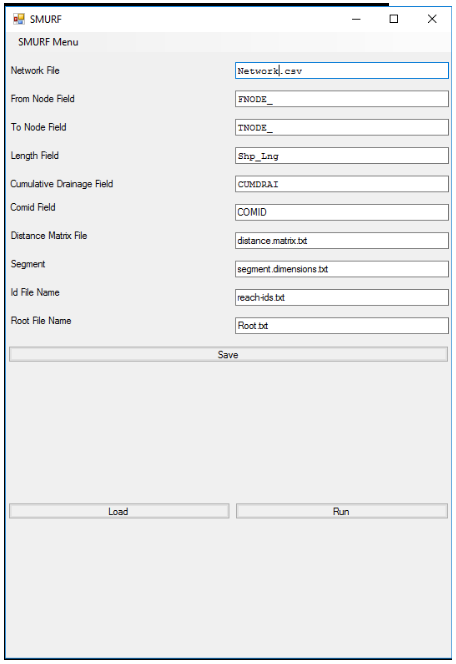
There are multiple processes you can use to run the SMRF model. The easiest to follow is the by using the Graphic User Interface, or GUI, which is in `SMRF/bin/SMRFGui.exe`. The following images and description will guide you through the steps necessary to run a model in the correct order.

**WELCOME SCREEN**



Simulating Meta-communities of Riverine Fishes

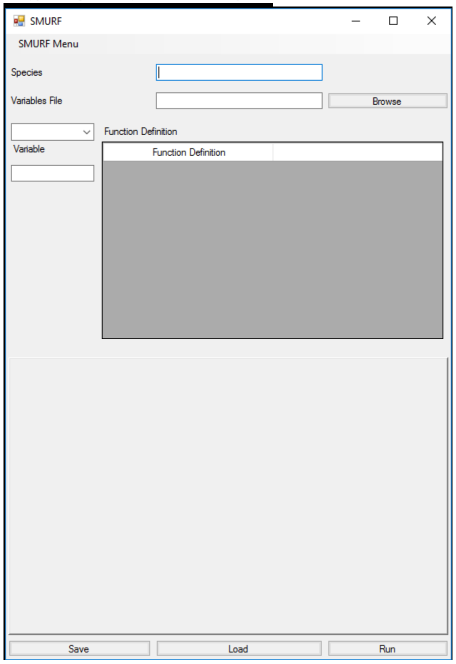
**NETWORK GENERATOR**



| | |
|---|---|
| Network File | Network.csv |
| From Node Field | FNODE_ |
| To Node Field | TNODE_ |
| Length Field | Shp_Lng |
| Cumulative Drainage Field | CUMDRAI |
| Comid Field | COMID |
| Distance Matrix File | distance.matrix.txt |
| Segment | segment.dimensions.txt |
| Id File Name | reach-ids.txt |
| Root File Name | Root.txt |

*The Network Generator* is used to customize and create the network files for the SMRF model. This uses the file `distance.xml` as input, and generates a `.csv` file containing the network information, and additional reference files. These files are references by multiple other files later in the model run, so this step must be done first.

## HSI GENERATOR



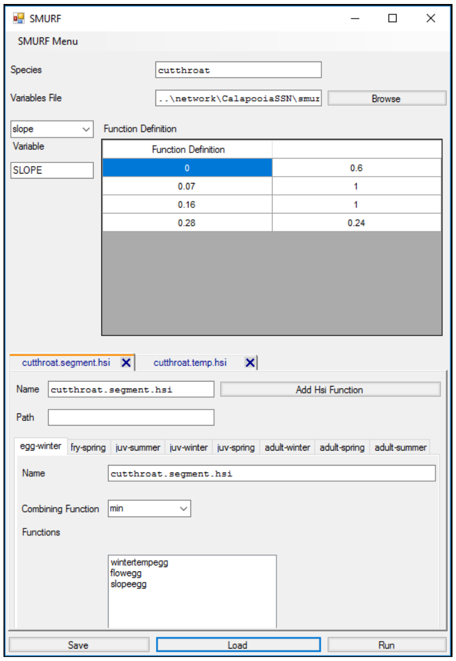
Species

Variables File

Function Definition

Variable

| Function Definition | |
|---|---|



Species: cutthroat

Variables File: ..\network\CalapooiaSSN\smur

Variable: slope

SLOPE

| Function Definition | |
|---|---|
| 0 | 0.6 |
| 0.07 | 1 |
| 0.16 | 1 |
| 0.28 | 0.24 |

cutthroat.segment.hsi | cutthroat.temp.hsi

Name: cutthroat.segment.hsi

Path

egg-winter | fry-spring | juv-summer | juv-winter | juv-spring | adult-winter | adult-spring | adult-summer

Name: cutthroat.segment.hsi

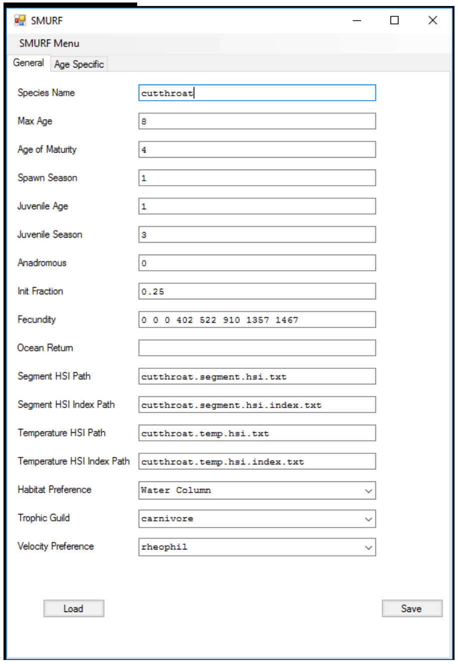
Combining Function: min

Functions:

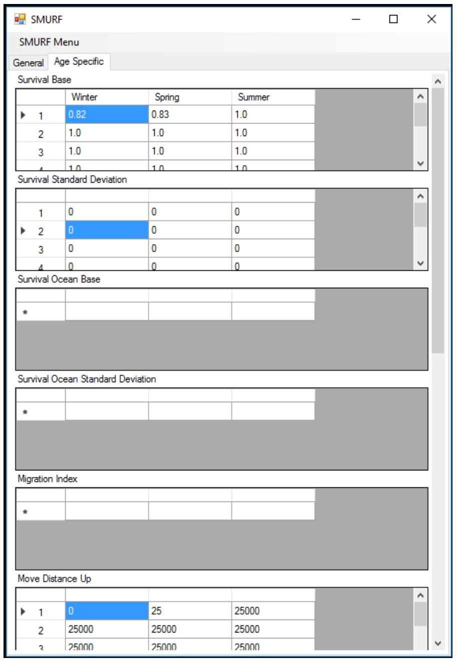
wintertempegg
flowegg
slopeegg

*The HSI Generator* produces the reference Habitat Suitability Index files from the input file `species.hsi.description.xml`. When a description is loaded, the window shows you all of the HSI function sets for every season and age class for the species. After loading, click *Run* to produce the HSI files that get referenced during the model run.

## SPECIES



| | |
|---|---|
| Species Name | cutthroat |
| Max Age | 8 |
| Age of Maturity | 4 |
| Spawn Season | 1 |
| Juvenile Age | 1 |
| Juvenile Season | 3 |
| Anadromous | 0 |
| Init Fraction | 0.25 |
| Fecundity | 0 0 0 402 522 910 1357 1467 |
| Ocean Return | |
| Segment HSI Path | cutthroat.segment.hsi.txt |
| Segment HSI Index Path | cutthroat.segment.hsi.index.txt |
| Temperature HSI Path | cutthroat.temp.hsi.txt |
| Temperature HSI Index Path | cutthroat.temp.hsi.index.txt |
| Habitat Preference | Water Column |
| Trophic Guild | carnivore |
| Velocity Preference | rheophil |



Survival Base

| | Winter | Spring | Summer |
|---|---|---|---|
| 1 | 0.82 | 0.83 | 1.0 |
| 2 | 1.0 | 1.0 | 1.0 |
| 3 | 1.0 | 1.0 | 1.0 |
| 4 | 1.0 | 1.0 | 1.0 |

Survival Standard Deviation

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 |

Survival Ocean Base

| | | | |
|---|---|---|---|
| * | | | |

Survival Ocean Standard Deviation

| | | | |
|---|---|---|---|
| * | | | |

Migration Index

| | | | |
|---|---|---|---|
| * | | | |

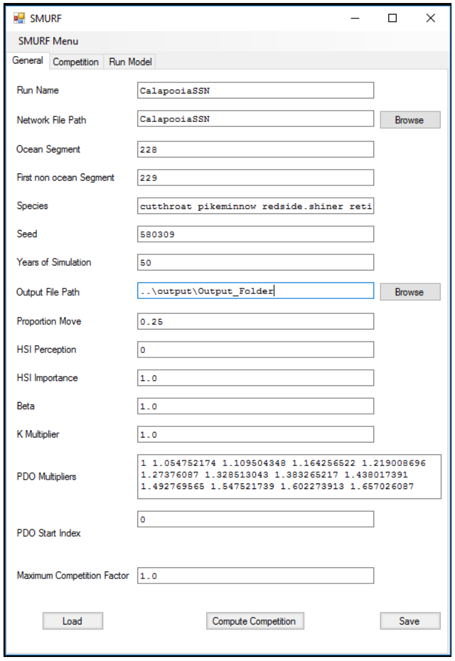
Move Distance Up

| | | | |
|---|---|---|---|
| 1 | 0 | 25 | 25000 |
| 2 | 25000 | 25000 | 25000 |
| 3 | 25000 | 25000 | 25000 |

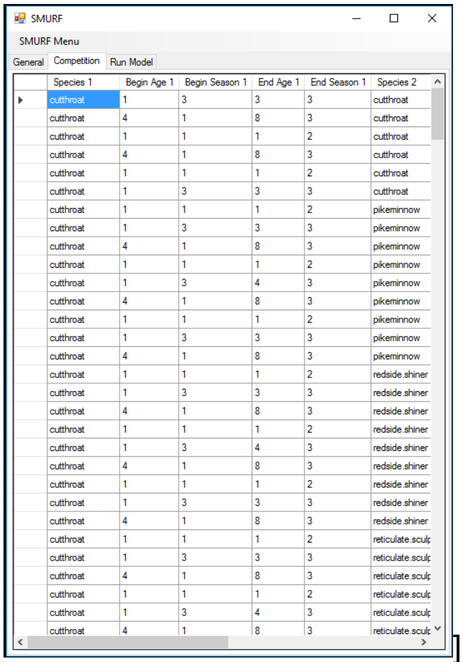
*Species* allows you to see and change any information about how a specific species operates. In the first tab *General* there are the species' life stages, fecundity, and general habitat preferences. In the *Age*

*Specific* tab there are matrices that reflect the survival, move distance, and other factors based on age and season. Unless you want to purposefully run a model with changes to species attributes, you do not need to edit these files. If you do choose to edit, make sure that you have an unaltered copy of the `species.xml` saved in a different folder so you can revert to the default values.

## RUN



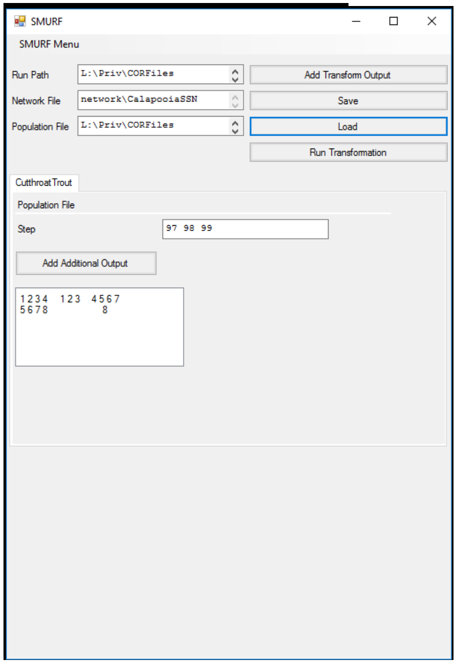
| | |
|---|---|
| Run Name | CalapooiaSSN |
| Network File Path | CalapooiaSSN |
| Ocean Segment | 228 |
| First non ocean Segment | 229 |
| Species | cutthroat pikeminnow redside.shiner reti |
| Seed | 580309 |
| Years of Simulation | 50 |
| Output File Path | ..\output\Output_Folder |
| Proportion Move | 0.25 |
| HSI Perception | 0 |
| HSI Importance | 1.0 |
| Beta | 1.0 |
| K Multiplier | 1.0 |
| PDO Multipliers | 1 1.054752174 1.109504348 1.164256522 1.219008696 1.27376087 1.328513043 1.383265217 1.438017391 1.492769565 1.547521739 1.602273913 1.657026087 |
| PDO Start Index | 0 |
| Maximum Competition Factor | 1.0 |



| Species 1 | Begin Age 1 | Begin Season 1 | End Age 1 | End Season 1 | Species 2 |
|---|---|---|---|---|---|
| cutthroat | 1 | 3 | 3 | 3 | cutthroat |
| cutthroat | 4 | 1 | 8 | 3 | cutthroat |
| cutthroat | 1 | 1 | 1 | 2 | cutthroat |
| cutthroat | 4 | 1 | 8 | 3 | cutthroat |
| cutthroat | 1 | 1 | 1 | 2 | cutthroat |
| cutthroat | 1 | 3 | 3 | 3 | cutthroat |
| cutthroat | 1 | 1 | 1 | 2 | pikeminnow |
| cutthroat | 1 | 3 | 3 | 3 | pikeminnow |
| cutthroat | 4 | 1 | 8 | 3 | pikeminnow |
| cutthroat | 1 | 1 | 1 | 2 | pikeminnow |
| cutthroat | 1 | 3 | 4 | 3 | pikeminnow |
| cutthroat | 4 | 1 | 8 | 3 | pikeminnow |
| cutthroat | 1 | 1 | 1 | 2 | pikeminnow |
| cutthroat | 1 | 3 | 3 | 3 | pikeminnow |
| cutthroat | 4 | 1 | 8 | 3 | pikeminnow |
| cutthroat | 1 | 1 | 1 | 2 | redside.shiner |
| cutthroat | 1 | 3 | 3 | 3 | redside.shiner |
| cutthroat | 4 | 1 | 8 | 3 | redside.shiner |
| cutthroat | 1 | 1 | 1 | 2 | redside.shiner |
| cutthroat | 1 | 3 | 4 | 3 | redside.shiner |
| cutthroat | 4 | 1 | 8 | 3 | redside.shiner |
| cutthroat | 1 | 1 | 1 | 2 | redside.shiner |
| cutthroat | 1 | 3 | 3 | 3 | redside.shiner |
| cutthroat | 4 | 1 | 8 | 3 | redside.shiner |
| cutthroat | 1 | 1 | 1 | 2 | reticulate.sculp |
| cutthroat | 1 | 3 | 3 | 3 | reticulate.sculp |
| cutthroat | 4 | 1 | 8 | 3 | reticulate.sculp |
| cutthroat | 1 | 1 | 1 | 2 | reticulate.sculp |
| cutthroat | 1 | 3 | 4 | 3 | reticulate.sculp |
| cutthroat | 4 | 1 | 8 | 3 | reticulate.sculp |

*Run* is where all of the mechanics of the model run are assessed, including the species involved in the run, HSI perception, and competition. After loading in a `Run.xml` file on the *General* tab, you can set your desired output folder and customize the present values. You can then select the *Compute Competition* button and follow the prompts to select the correct species. After computing competition, values representing the competition become visible on the *Competition* tab. To cache any changes made to the run file, you must **save the file again before running the model**. Once that is done you can navigate to the *Run Model* tab and select *Run* to begin.

## TRANSFORM OUTPUT



Run Path: L:\Priv\CORFiles

Network File: network\CalapooiaSSN

Population File: L:\Priv\CORFiles

Step: 97 98 99

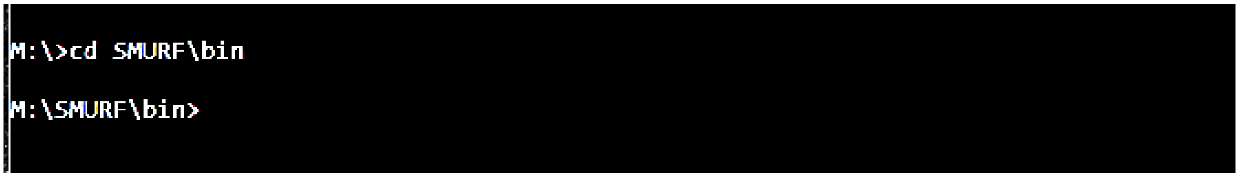
```
1234  123  4567
5678         8
```

*Transform Output* lets you take the raw output files from a model run and change it to a more understandable format. After loading a `transform.xml` file, you can see the pathing to that file's SMRF Directory, network folder, and raw output. The tab below is named the same as the file that the transform will yield. You can also change the time steps of the model incorporated in the transform and add additional output fields that are the sum of specific ages. Similar to altering *Run* files, **all changes must be saved before you run the transform**.

## VIII) RUNNING SMRF ON THE COMMAND LINE

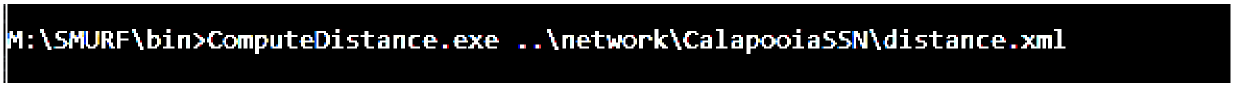
The other way of running the SMRF model is by using the Command Line. This process is less intuitive than the GUI, because the files and their contents are not immediately visible or editable. It also requires that the files be prepared in advance, which in the GUI you can create and save some files from scratch. However, if you are familiar with the Command Line you can run models very efficiently using this method. The screenshots found in this section are to provide examples of commands to run the model, each command must be altered to fit your personal network.

To start, you must open you Command Prompt (`cmd.exe`). Then use the `cd` command to change your working directory to the `SMRF\bin` folder. Notice that Command Line used back-slashes and will not recognize forward-slashes in pathing.



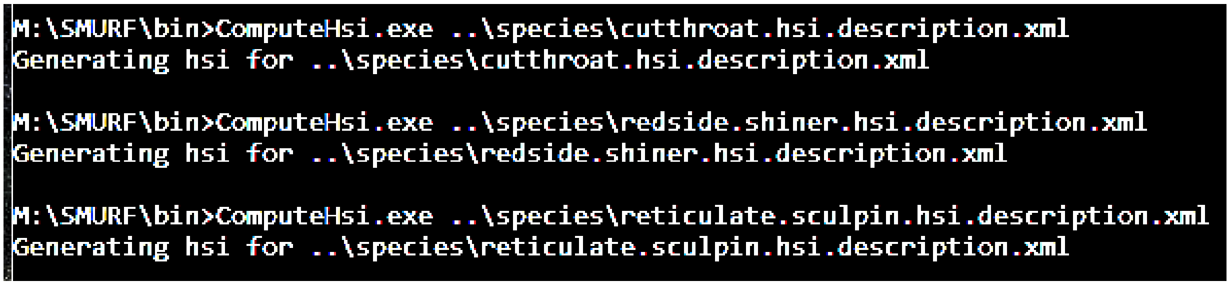
```
M:\>cd SMURF\bin

M:\SMURF\bin>
```

For the next few lines, you need to call specific executable `(.exe)` files in the `bin` folder, then follow it up by pathing to the `.xml` files that the executable requires. The first of these is computing the auxiliary network files from `distance.xml`. To do this you must call the executable `ComputeDistance.exe`, followed by pathing to `distance.xml`. Since this file is in the `network` folder, you must the "..\" notation to back out of the `bin` folder.



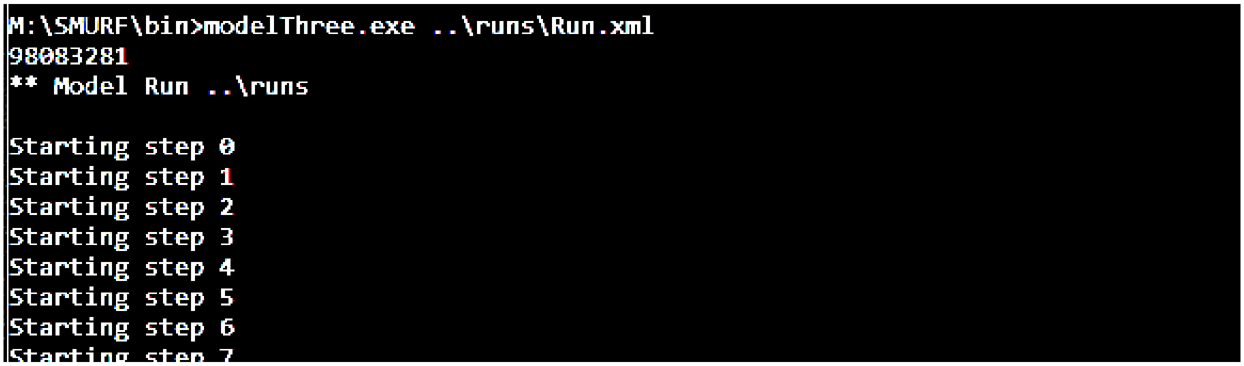
```
M:\SMURF\bin>ComputeDistance.exe ..\network\CalapooiaSSN\distance.xml
```

The same command structure is then used to compute HSI by calling `ComputeHsi.exe`, followed by a species' `species.hsi.description.xml`. Note that even if you have multiple species to prepare the HSI descriptions for, you need to call and execute each HSI description individually. After executing the command, there will be a short response acknowledging that the HSI is being generated.



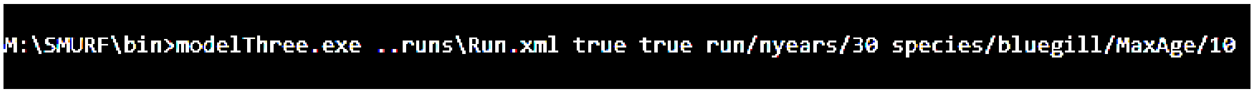
```
M:\SMURF\bin>ComputeHsi.exe ..\species\cutthroat.hsi.description.xml
Generating hsi for ..\species\cutthroat.hsi.description.xml

M:\SMURF\bin>ComputeHsi.exe ..\species\redside.shiner.hsi.description.xml
Generating hsi for ..\species\redside.shiner.hsi.description.xml

M:\SMURF\bin>ComputeHsi.exe ..\species\reticulate.sculpin.hsi.description.xml
Generating hsi for ..\species\reticulate.sculpin.hsi.description.xml
```

The next step is to run the model. This incorporates the same command structure as the last two steps, this time by calling `modelThree.exe` and pathing to a `Run.xml` file. During this step, you'll notice an important difference between the GUI run and the Command Line run: The Command Line has no option to compute competition. Since this has the potential to impact the accuracy of the results, it is recommended that most runs be done in the GUI. However, if you have a run file that had its competition computed in the GUI and then re-saved in the file, this file can still be run on the Command Line with the competition included.



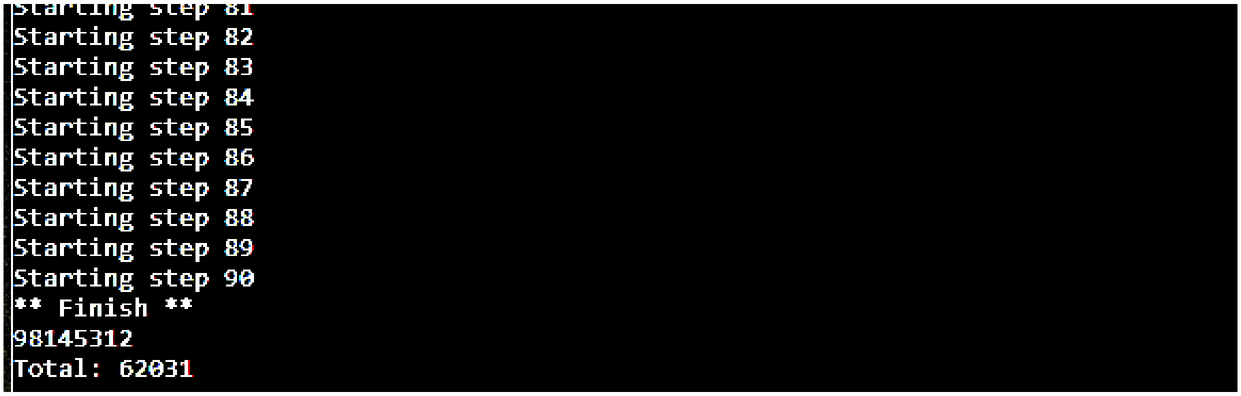
```
M:\SMURF\bin>modelThree.exe ..\runs\Run.xml
98083281
** Model Run ..\runs

Starting step 0
Starting step 1
Starting step 2
Starting step 3
Starting step 4
Starting step 5
Starting step 6
Starting step 7
```

Additional arguments can be included after the run file reference to change the parameters of the run. The next two arguments are logical and determine whether you want to apply movement or survival values to fish being recruited at each step, respectively. These values default to "true." The following steps allow you to adjust the values in a species or run file without opening and resaving them. Multiple changes can be applied, one after the other, and they are executed by inputting the keyword "run" or species", pathing to the attribute, and writing a new value or values.



```
M:\SMURF\bin>modelThree.exe ..runs\Run.xml true true run/nyears/30 species/bluegill/MaxAge/10
```

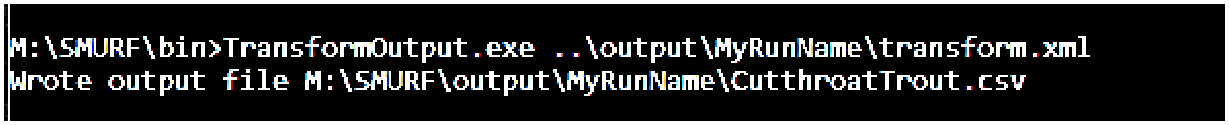
When you execute this function, it displays the same text that the GUI shows when running the model. The initial number (98093281 in this case) is a random seed for the run, and the "Starting Step" text shows that the model in processing a specific time step. Each step represents one of the seasons for a given year (Winter, Spring, and Summer). In this example, the model is being run for 30 years, so the model ends after time step 90.



```
Starting step 81
Starting step 82
Starting step 83
Starting step 84
Starting step 85
Starting step 86
Starting step 87
Starting step 88
Starting step 89
Starting step 90
** Finish **
98145312
Total: 62031
```

For more information on the records created for a particular model run, check the `log.txt` file produced in the output folder. More information about this file can be found in section XII.

The final step of this model run is transforming the output. You do this by calling `TransformOutput.exe` followed by the pathing to the desired `transform.xml` file. It is recommended that this file get saved in the output folder created by the last step. If the transform is successful, the Command Prompt will inform you.



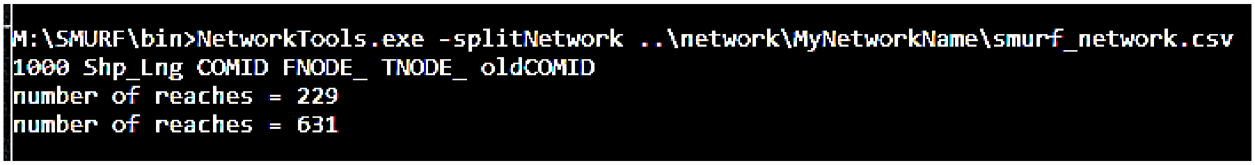
```
M:\SMURF\bin>TransformOutput.exe ..\output\MyRunName\transform.xml
Wrote output file M:\SMURF\output\MyRunName\CutthroatTrout.csv
```

## IX) THE NETWORK-SPLITTING TOOL

When a user is preparing the necessary data requirement to run the SMRF model, particularly the network `.csv` or `.dbf`, it is possible to have a stream reach that has a longer length than species have Move Distance values. This creates a major problem when running the model, because fish are able to enter this stream reach but do not have the Move Distance to leave. In order to combat this, you can utilize the application in the `SMRF/bin` folder: `NetworkTools.exe`. This section will outline the steps that must be carefully taken to disassemble the network, run the model on the shortened reaches, and reassemble everything.

The first step for this process is to create a copy of the folder that your network files are located in to prevent any irreparable damage to the original network files. This should be a folder nested in the `SMRF/network` folder. Once you copy the copy of the network, ensure that the new folder has a copy of original `distance.xml` file, and the network file in the format of a `.csv` or `.dbf`. If it is in `.dbf` format, also copy the `.prj`, `.shp`, and `.shx` files, and then re-save the `.dbf` as a `.csv` for compatibility with R.

The next step is to open the Command Prompt, set your directory to the `SMRF/bin` folder, and call `NetworkTools.exe`. See section IX for directions on how to run the model from Command Prompt. You must follow up the application with several exact specifications. The screenshot below is exactly what must be typed, followed by what each of these represents.



```
M:\SMURF\bin>NetworkTools.exe -splitNetwork ..\network\MyNetworkName\smurf_network.csv
1000 Shp_Lng COMID FNODE_ TNODE_ oldCOMID
number of reaches = 229
number of reaches = 631
```

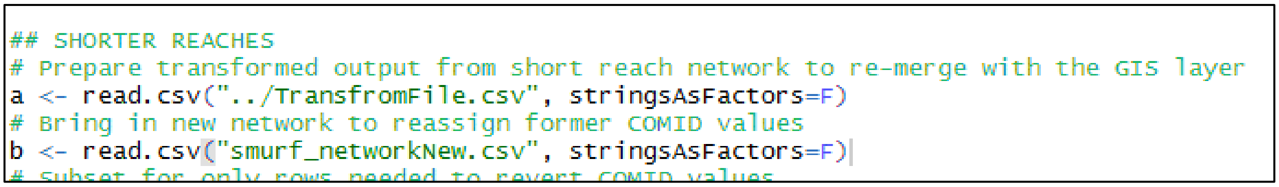
- `-splitNetwork`
    - This call is to tell the Command Prompt that you are using this function to split the network.
- `..\network\MyNetwork\smrf_network.csv`
    - This is the pathing to the network that you want to split the reaches of. This may differ dependent on what you network name/location is.
- `1000`
    - This is a suggestion for the desired length of the of the reaches in the new network. When a reach needs to be split, its length is divided by this number and rounded to the nearest integer, then the reach is split into that many equally sized reaches.
- `Shp_Lng COMID FNODE_ TNODE_ oldCOMID`
    - The Network Splitting Tool ends up adding and changing multiple columns in the network file. These entries provide header names for these columns and must be listed in this order. It is recommended that you keep this naming convention, but if you have your own for one of these columns, make sure you follow this pattern.

Two lines of response appear when this is entered correctly. The first shows the number of reaches present in the original network, the second shows how many reaches are in the newly created network. This new network will appear in the same folder as the old network as a `.csv` with "New" appended to the end of the name.

Next, you need stay on the Command Line and recalculate the distance files and the HSI files. However, **before you begin to compute**, you must open up HSI Description files as well as the `distance.xml` and change your network pathing to reflect the newly created network instead of the old one. This pathing affects the files that are generated, so they must be regenerated to ensure the HSI and network files will work on this new network. After these are generated, switch to the GUI to set up the `Run.xml` and to compute competition, then save the file and run the model. See section VII for an example on how to run the model using the GUI. Once the model has finished running, you can set up a transform file (that must correctly path to the new network) and transform the output using wither the GUI or the Command Line.

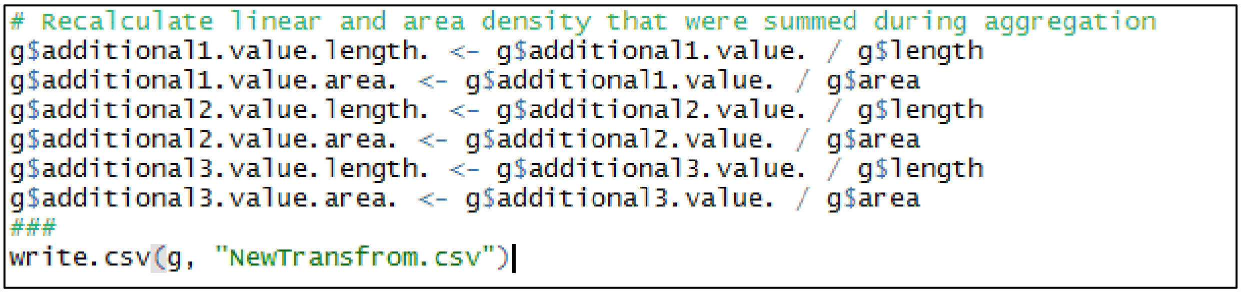
Now that we have a transformed output for the split network, we must keep the output while piecing the network back together. To do this, refer to the R Script found in the Documentation folder, named `CleanedMapScript.R`. Before running any of the script though, you must open the new transform file that you have created. If your network is prepared to handle anadromous species, then there will be segments in the transform files that represent ocean connecters. These must be deleted from this transform file, or the R Script will result in an error.

In the Script, you will first change the `setwd` command to reflect the location of your network, then you will load all the packages listed. The next block of code should be labeled `SHORTER REACHES`.



```
## SHORTER REACHES
# Prepare transformed output from short reach network to re-merge with the GIS layer
a <- read.csv("../TransfromFile.csv", stringsAsFactors=F)
# Bring in new network to reassign former COMID values
b <- read.csv("smurf_networkNew.csv", stringsAsFactors=F)
# Subset for only rows needed to revert COMID values
```

The first pathing reference in this block should be altered to direct to your transformed output file. Keep in mind that you have already changed your working directory, so this pathing will start from the location of your network. The next pathing line should identify the name of your split network. After changing these two lines, you will then run the remained of this block of code in sequential order, ending with this:



```
# Recalculate linear and area density that were summed during aggregation
g$additional1.value.length. <- g$additional1.value. / g$length
g$additional1.value.area. <- g$additional1.value. / g$area
g$additional2.value.length. <- g$additional2.value. / g$length
g$additional2.value.area. <- g$additional2.value. / g$area
g$additional3.value.length. <- g$additional3.value. / g$length
g$additional3.value.area. <- g$additional3.value. / g$area
###
write.csv(g, "NewTransfrom.csv")
```

The last command, `write.csv`, is creating the finalized transform file fit to the original network. The name can be customized to any desired convention but must include the file extension.

## X) INTERMEDIATE SMRF OUTPUT (Produced by Network and HSI Generators)

Below is a description of the different files produced by SMRF during model preparation.  This includes network geometry files generated by ComputeDistance.exe in the Network Generator and species-specific HSI files generated by ComputeHSI.exe in the HSI Generator.  Users should not modify these files.  They will be required to successfully run a SMRF model.

NETWORK GENERATOR

- `distance.matrix.txt` – contains a matrix with unique stream reaches as both rows and columns.  Used for movement within the model and indicates the distance between each stream reach, much like the distance chart in a road atlas.
- `reach-ids.txt` – contains unique reach ID numbers.  Each row represents a unique stream reach
- `Root.txt` – contains a single value indicating the root of the stream network tree.
- `segment.dimensions.txt` – contains geometry for each unique stream segment.  Columns are segment width (meters), length (meters), and area (m^2) from left to right.  Each row represents a river segment, which corresponds by row order to the IDs contained in `reach.ids.txt`.

HSI GENERATOR

*Note:  other than correctly specifying the names of these output files in the HSI Generator, and properly referencing them in the species.xml file, there should be no need for users to directly work with or modify these files.  What follows is a description of file contents to assist users with interpretation if they choose to inspect this intermediate model output.*

- *`Species`*`.segment.hsi.index.txt` – identifies the function determining HSI (slope and flow) for each time step in a species' life-history.  Rows represent age-class (years) and columns represent annual time-steps. Values reflect the order that the function appears in the *`species`*`.segment.hsi` function code of the *`species`*`.hsi.description.xml` file.  For example, if there are 9 functions in this section of code, the fifth function occurring in the sequence would be indexed as 5.

- `Species.segment.hsi.txt` – contains a table of species-specific HSI values ranging between 0.0 and 1.0 for each stream reach and season/stage in a species life history. Rows are unique stream reaches and columns are HSI function sets ordered as they appear in the `species.segment.hsi` function code of the `species.hsi.description.xml` file. Each column reflects the final score assignment for the reach based on the lowest parameter HSI value (minimum of slope, flow, and temperature HSI values).
- `Species.segment.hsi.limit.txt` – These files are not referenced by the model but allow users to diagnose which of the HSI functions (temperature, slope, or flow) is limiting habitat use in a given reach. Rows are stream reaches, columns are **HSI FUNCTION SETS** in the order found under the `species.segment.hsi` code in the `species.hsi.description.xml` file. Values indicate the limiting function (minimum score) within the corresponding function set starting with 0 (0 = first function listed in the function set, 1 = second function listed in the function set, etc.)
- `Species.temp.hsi.index.txt` – identifies the function determining HSI (temperature) for each time step in a species' life-history. Rows represent age-class (years) and columns represent annual time-steps. Values reflect the order that the function appears in the `species.temp.hsi` function code of the `species.hsi.description.xml` file. For example, if there are 9 functions in this section of code, the fifth function occurring in the sequence would be indexed as 5.
- `Species.temp.hsi.txt` – species-specific values for each HSI function (temperature). Rows are unique stream reaches and columns are individual HSI functions ordered as they appear in the `species.temp.hsi` function code of the `species.hsi.description.xml` file. Values are HSI scores ranging between 0.0 and 1.0.
- `Species.temp.hsi.limit.txt` – These files are not referenced by the model but allow users to diagnose which of the HSI functions is limiting habitat use in a given reach. Rows are stream reaches, columns are **HSI function sets** in the order found in the `species.hsi.description.xml` file. Values reflect the limiting function (minimum score) within the corresponding function set, starting with 0 (0 = first function listed in the function set, 1 = second function in the function set, etc.)
    - Contains duplicate information for flow/slope also found in the `species.segment.hsi.limit.txt` file. `segment.hsi.limit` entries are repeated (see previous bullet point), with `temp.hsi.limit` values for the `temp.hsi` function sets appended at the end of the file.

## XI) DESCRIPTION OF MODEL RUN OUTPUT (Contents/Interpretation)

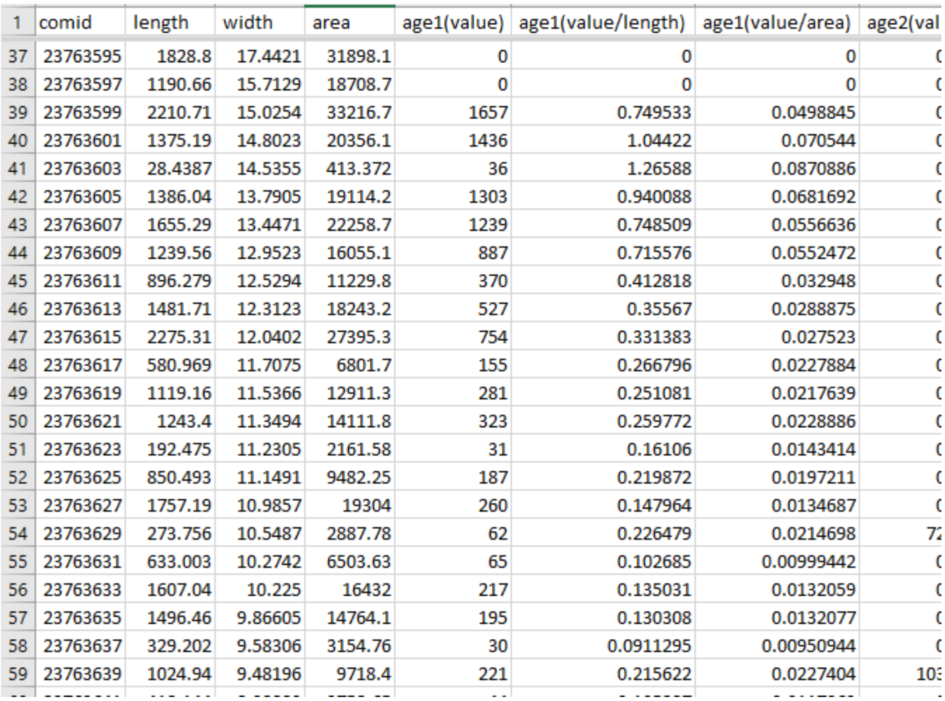
RAW OUTPUT

- Running the model produces various types of output files:
    - `log.txt` – log file recording life history events (movement, survival, recruitment, promotion) at each time-step of the model run. Contains entries for each species/age-class included in modeled fish assemblage.
    - *species*`Pops.txt` – population files, one file for each species containing the number of individuals of each age at each reach for each time step;
    - *species*`PopTotals.txt` – model run summary showing system wide species abundances (total population) by age-class
- Format the population file (*species*`Pops.txt`) as desired using the Transform Output feature of the GUI or running TransformOutput.exe on the command line (described below)

TRANSFORMED OUTPUT

Transform files reformat the output model data into a more readable format (.csv) and summarize age-classes and seasons (steps). Each row of this csv represents a unique reach of the stream network, organized by the first column: comid. Though this field header is lowercase, it corresponds fully to the COMID field found in the `network.csv` field. The following 3 columns (length, width, area) describe that particular stream reach, denoting its length (m), average width (m), and total area covered ($m^2$).

Every column following these first 4 is organized into sets of three, with each new set representing a different age class for the species. The first of the three shows the total abundance of fish that are a specific reach for that time step, or the average amount if multiple time steps are specified. The second and third divide the total abundance by that reach's length and area values, respectively, to calculate that reach's linear density and area density. For example, the head for the first set of three read as such: age1(value), age1(value/length), age1(value/area). If desired, these headers can be renamed to reflect the specific units: Fish, Fish/m, and Fish/$m^2$). These triplets repeat for every age class observed by the species. The columns that follow the age classes may or may not be present, depending on whether the `transform.xml` included additional outputs. For each additional output, there will be a new set of three columns that show to total abundance, linear density, and area density for all age classes listed in that additional output. The picture below is an example of a transformed output `.csv`, including the starting reach-data columns and abundances for the species at age 1.



| 1 | comid | length | width | area | age1(value) | age1(value/length) | age1(value/area) | age2(val |
|---|---|---|---|---|---|---|---|---|
| 37 | 23763595 | 1828.8 | 17.4421 | 31898.1 | 0 | 0 | 0 | 0 |
| 38 | 23763597 | 1190.66 | 15.7129 | 18708.7 | 0 | 0 | 0 | 0 |
| 39 | 23763599 | 2210.71 | 15.0254 | 33216.7 | 1657 | 0.749533 | 0.0498845 | 0 |
| 40 | 23763601 | 1375.19 | 14.8023 | 20356.1 | 1436 | 1.04422 | 0.070544 | 0 |
| 41 | 23763603 | 28.4387 | 14.5355 | 413.372 | 36 | 1.26588 | 0.0870886 | 0 |
| 42 | 23763605 | 1386.04 | 13.7905 | 19114.2 | 1303 | 0.940088 | 0.0681692 | 0 |
| 43 | 23763607 | 1655.29 | 13.4471 | 22258.7 | 1239 | 0.748509 | 0.0556636 | 0 |
| 44 | 23763609 | 1239.56 | 12.9523 | 16055.1 | 887 | 0.715576 | 0.0552472 | 0 |
| 45 | 23763611 | 896.279 | 12.5294 | 11229.8 | 370 | 0.412818 | 0.032948 | 0 |
| 46 | 23763613 | 1481.71 | 12.3123 | 18243.2 | 527 | 0.35567 | 0.0288875 | 0 |
| 47 | 23763615 | 2275.31 | 12.0402 | 27395.3 | 754 | 0.331383 | 0.027523 | 0 |
| 48 | 23763617 | 580.969 | 11.7075 | 6801.7 | 155 | 0.266796 | 0.0227884 | 0 |
| 49 | 23763619 | 1119.16 | 11.5366 | 12911.3 | 281 | 0.251081 | 0.0217639 | 0 |
| 50 | 23763621 | 1243.4 | 11.3494 | 14111.8 | 323 | 0.259772 | 0.0228886 | 0 |
| 51 | 23763623 | 192.475 | 11.2305 | 2161.58 | 31 | 0.16106 | 0.0143414 | 0 |
| 52 | 23763625 | 850.493 | 11.1491 | 9482.25 | 187 | 0.219872 | 0.0197211 | 0 |
| 53 | 23763627 | 1757.19 | 10.9857 | 19304 | 260 | 0.147964 | 0.0134687 | 0 |
| 54 | 23763629 | 273.756 | 10.5487 | 2887.78 | 62 | 0.226479 | 0.0214698 | 72 |
| 55 | 23763631 | 633.003 | 10.2742 | 6503.63 | 65 | 0.102685 | 0.00999442 | 0 |
| 56 | 23763633 | 1607.04 | 10.225 | 16432 | 217 | 0.135031 | 0.0132059 | 0 |
| 57 | 23763635 | 1496.46 | 9.86605 | 14764.1 | 195 | 0.130308 | 0.0132077 | 0 |
| 58 | 23763637 | 329.202 | 9.58306 | 3154.76 | 30 | 0.0911295 | 0.00950944 | 0 |
| 59 | 23763639 | 1024.94 | 9.48196 | 9718.4 | 221 | 0.215622 | 0.0227404 | 103 |

The structure of the transformed output being organized by stream reach makes it the perfect format for plotting using GIS software. On any GIS software (ArcGIS, QGIS, etc.), you can import the transformed output and the network shapefile that the model was run on. These two files share identical `comid` and `COMID` columns, which allows you to join the two tables by these values. Once joined you can customize the coloration of the shapefile based on any of the data present in the transformed output, making it ideal for comprehending and present conclusions form the model.

## XII) CALIBRATING THE MODEL

The default process for establishing SMRF parameters is to collect literature-backed values for each species' parameters and compile them into the appropriate .xml files. However, due to the complexity of fish movement, growth, and interactions, calibration will be required to tune the model into satisfactory behavior.

Calibration is a process that is used to identify the optimal values for each parameter of each species. Calibration presumes existence of fish assemblage sampling data for locations within the modeled stream network. By making a variety of changes to the values of each parameter (within the accepted ranges provided by literature) it is possible to identify the values that produce the least amount of error when comparing the modeled data to observed data. Once identified, the default values can be permanently changed to reflect the optimal points, improving model performance.

### METHODS

The calibration process starts with a sensitivity analysis of all species and parameters (see *Sensitivity Appendix)*. In addition to identifying portions of the model that may have problems, the Sensitivity Analysis also reveals which parameters have the largest effect on the relative abundance and percent occupancy of each species. Parameters that have the greatest effect are the most likely to cause higher errors, because a small adjustment in the parameter can lead to the biggest change in the metrics. Highly sensitive parameters are prioritized during calibration.

Potential calibration values for each parameter need to be within acceptable limits and not arbitrary. The minimum and maximum values are determined from the literature or expert judgement where data are lacking. Once these minimum and maximum values have been identified, they are converted in terms of *proportional change from the default value* so they can be used in the scripts.

**CALIBRATION SCRIPTS**

There are 3 scripts that are used in calibration, each serving a specific function when executed sequentially. Annotated versions of all three are available in *SMRF/Documentation/Calibration.* The first takes the compiled data file of all fish sampling events, aggregates them to get the total number of each species found in each sampled reach, and separates the data into two files: the calibration subset and the validation subset. Each of these contain subset of reach sampling data, 70% of them for calibrating the model output and 30% for confirming the validity of these calibrations.

The second script is where the calibration changes are made, the SMRF runs are performed, and the output is calculated and organized into csv files. The script loads in the network and calibration files, prepares functions to edit the values of the chosen parameters, and generates random proportional change values within the literature-defined range of each parameter. Unlike the sensitivity analysis script, which generates separate vectors of proportional change values for each parameter, the calibration script organized the vectors into a table so that every SMRF run includes changes to each of the parameters in question. Once this is set up, the appropriate species and run files are updated and the SMRF runs are executed.

The bulk of this script comes after the SMRF runs, when the outputs are being compiled, calculated, and analyzed. First, many empty vectors are created, each of them to be filled in with calculated output over the course of the following loop. The vectors will store the data for the relative abundance, occupancy, and total abundance for each species within the reach subset, as well as the error calculations for each of these values. The loop then begins, starting the same way as the sensitivity's does by reading in and transforming the population files from all the SMRF runs. The final vectors for each species, depicting their total abundance in each reach, are then bound into a single data frame. This data frame is merged with the calibration subset, limiting the scope of reaches only to those that were sampled.

From here on, the script is comparing the calibration data to the SMRF-predicted data and calculating the Root-Mean Squared Error (RMSE) between the two in multiple categories. The formula for RMSE can be found in the equation below. The first calculations are for percent occupancy RMSE, comparing the percentage of sampled reaches that contain a certain species of fish to that fish's occupancy in the same reaches in SMRF. For the purposes of these calculation, occupancy is simplified as a binary where any reach with over .5 fish is rounded to 1, and any with less are rounded to 0.

$$RMSE = \surd(sum((predicted - observed)^2))$$

The following calculations, for relative abundance RMSE, require more steps that the percent occupancy. First, all the abundances for both the calibration subset and SMRF predictions are summed separately. Using these sums, the relative abundances of a species' observed and modeled data in each reach is calculated. Prior to the calculation of RMSE, the entire data set is subset to only include reaches that have abundance for both observed and predicted so that the vast amount of reaches without data in one or the other category would not inflate the error. Then the RMSE is calculated between the predicted and observed relative abundance.

In addition to these RMSE calculations, the RMSE for total model abundance is calculated as well. Supplemental information is then summarized, namely the total predicted abundances for each species and their total occupant per reach. After this loop finishes, all the output vectors are bound into 2 data frames: one for all the RMSE values, and one for the raw data that the RMSEs was calculated from. Another 2 columns are added to the RMSE data frame that sum the RMSEs of each category to get the total relative abundance RMSE and total percent occupancy RMSE. Each final output table is then written out as a csv to be used in the final calibration script.

The function of the last script is to read in the second script's `csv's` and write out all figures to explain the output. The figures describe how the Relative Abundance, Occupancy, and Total Abundance of each species reacted to each parameter's change. In addition, the figures describe the changes in the RMSEs and the total RMSEs. Adjustments can then be made and evaluated via an iterative process of testing and evaluation model outputs until satisfactory model performance in achieved.

# XIII) TROUBLESHOOTING AND FREQUENTLY ASKED QUESTIONS

## Why is my SMRF Program not running correctly?

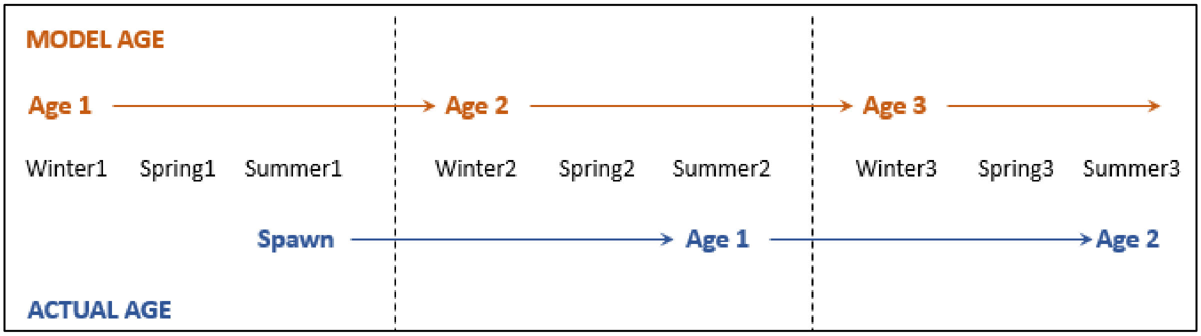
- All the SMRF executable files (`.exe`) have the potential to fail if the arguments are assigned incorrectly
    - In some cases, an error message will appear. Read this message to see if it provides context for the error. If not, refer to the general troubleshooting steps below.
- `NetworkTools.exe`
    - Check your network to make sure there are reaches long enough that the Network Splitting tool is applicable
    - Needs to be run using the command line, with the directory set to `SMRF/bin`
    - The commands and pathing must be typed and spelled correctly
    - New column headers but be spelled correctly and in the specified order
        - If you are using your own network or unique naming convention, the order must be maintained
- `ComputeDistance.exe`
    - The `distance.xml` being called must be in the `SMRF/network/[MyNetworkName]` folder
    - Ensure your Network File is pointing to the correct file
        - Check that the file is in the correct format (`.csv` or `.dbf`)
        - Make sure you are directed to the current version of the files (split network compared to original)
- `ComputeHsi.exe`
    - The species/hsi.description.xml files must be copied into the `SMRF/species` folder
    - Every copied species/hsi.description.xml must have the network pathing direct to the same network used with `ComputeDistance.exe`
- `modelThree.exe`
    - Network File Path must be the name of the network subfolder in which the `ComputeDistance.exe` output is stored
    - `Species.xml` files must be copied into the S`MRF/species` folder
        - `Species.xml` files with a "`MaxAge`" value inconsistent with the number of rows in their parameter matrices will result in `modelThree.exe` failing without an error code
    - Species names must be spelled correctly and separated by a single space
    - The correct files must be called when computing competition
    - The run file must be saved and re-loaded directly before running
- `TransformOutput.exe`
    - The pathing for the network, `popsCounts`, and output must be oriented and spelled correctly
        - Check to make sure there is no additional unwanted text hidden from view in the fields
    - Ensure that the timesteps and additional age output are applicable to the species and SMRF run
    - Make sure that the transform file is saved and re-loaded before execution

## How long does a typical SMRF Run take?

- The length of a SMRF run can vary wildly, being as quick as a minute and long as 10-15 minutes. This is primarily dependent on 3 attributes of the run:
    - The number of years the model is predicted
    - The amount of reaches in the network
    - How many species are being modeled
- The number of years is self-explanatory. Running the model for 100 years will take twice as long as 50 years. The total number of years does not affect the processing time of a single timestep
- Each species being added to a scenario increases the run time exponentially, because SMRF needs to process the competition of that species with all other species in the model, including itself
- The number for reaches in the model has a large effect as well, increasing the runtime exponentially. This is the primary drawback of the network-splitting tool, which creates many more reaches in a network and slows it down. Even considering this, use of the network tool is advised to ensure model accuracy

## How does Model Age of a species work with SMRF?

- In SMRF, the Model Age of a fish is different from how old the fish is. In SMRF, the fish starts at Age 1 during the first timestep (Winter) of the year that species first appears. This is true even if the species spawns in the Spring or Summer; in those cases, the fish will only be Age 1 for one or two timesteps. The fish then ages one year at every Winter timestep until it reaches MaxAge
- The following graphic depicts the difference between Model Age and Actual Age for a species that spawns in season 3 (Summer)



## Does SMRF make my computer run slowly?

- When running the model (`modelThree.exe`), SMRF can draw a lot of processing power. This can make it difficult to navigate through the GUI or the SMRF directory while SMRF is running
    - Closing excessive applications before running SMRF is not necessary, but may prevent a system slowdown
- Using high-end or external systems will allow SMRF to run without risking interference with other operations. It will also affect the runtime of the model itself (a scenario run on an external server will be much faster than on an average laptop).
    - To see how much of your processing power SMRF is using, open the Task Manager and locate the information on `modelThree.exe`

## What Network Attributes can SMRF use in a run?

- SMRF can assess the impact of any stressor in a network. The model testing and calibration were conducted while using seasonal temperature, cumulative drainage, and flow as the only network attributes. Users can add their own data to any network, allowing it to test the effects of pollution, land use, and any other factors
- To successfully add an attribute to a network, the user MUST have:
    - A method of classifying that attribute on a numerical scale
    - A value for that attribute for every reach in the network
        - May require multiple values if the attribute changes seasonally
    - HSI data for every species being run on this network

## How many years (or timesteps) should my SMRF run predict?

- SMRF is not designed to model fish species' distributions and populations in response to environmental changes over time. The data that SMRF is using is stagnant, meaning that the conditions of the model run do not change. Instead, SMRF takes a snapshot of the environmental conditions at one point in time and runs the model until it reaches equilibrium
    - Any timestep after this equilibrium is reach will serve as a good indicator of fish distributions and populations for the given network scenario
- After initial recruitment, the populations of all species in the assemblage will vary wildly for many timesteps. It normally takes as many as 30-40 timesteps for the populations to reach equilibrium. It is recommended that a scenario is run for as many timesteps as is required for that scenario to reach equilibrium
    - Once equilibrium is reached, the populations will be a consistent oscillating trend of seasonal abundances
    - The point of equilibrium can be estimated by looking at the `SpeciesPopsTotals.txt` file and seeing when the abundances level into a consistent trend. For a more precise estimate, graph the abundances versus the timestep to visualize the trend.

## How should my SMRF directory be organized?

- The most important part of the SMRF directory is that master folder needs to be named [`SMRF`]. Changing the name of this folder to anything else may result in the model breaking. In addition, it cannot be nested inside of another folder named [`SMRF`]; this confuses the pathing and breaks the model
- We recommend using a unique and original naming convention for every SMRF run. After a while of using SMRF, there may be a lot of output subfolders from different runs. Establishing a clear and concise naming practice is the best way to ensure that you don't mistake some outputs for others.

SC
IEN
CE

EPA
United States
Environmental Protection
Agency

Office of Research and
Development (8101R)
Washington, DC 20460