

# THESIS

## THE DEVELOPMENT OF A DECISION SUPPORT SYSTEM FOR CONCURRENTLY EVALUATING CHANGES IN INSTREAM AND FLOODPLAIN HABITATS CAUSED BY FLOW MODIFICATIONS

Submitted by

Elaina Passero

Department of Civil and Environmental Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2020

Master's Committee:

Advisor: Ryan Morrison

Pierre Julien  
Matthew Ross

Copyright by Elaina Passero 2020  
All Rights Reserved

## ABSTRACT

### THE DEVELOPMENT OF A DECISION SUPPORT SYSTEM FOR CONCURRENTLY EVALUATING CHANGES IN INSTREAM AND FLOODPLAIN HABITATS CAUSED BY FLOW MODIFICATIONS

Assessments of changes to riverine ecosystems due to flow modifications have historically focused on instream habitat. Thus, considerations of floodplain habitat have often been neglected in assessment tools, creating difficulties for understanding the comprehensive impacts of flow changes to both instream and floodplain environments. To support improved habitat management and protection of naturally variable flows, I developed a decision support system that evaluates both fish and vegetation habitat availability in alternative flow scenarios. This system uses the results of high resolution 2D hydrodynamic models to quantify and map suitable habitat for fish and floodplain vegetation at a range of discharges in a river reach. Depth, velocity, and substrate habitat preference information was used to determine available fish habitat at each modeled discharge. Vegetation habitat was quantified from logistic regression equations relating long-term habitat inundation patterns to probability of occurrence of vegetation.

I demonstrate the use of this tool on the Verde River in Arizona, USA. Habitat was evaluated for the historic flow record and two alternative flow scenarios: reduced high flows and reduced baseflows. The two scenarios were compared by evaluating changes in monthly and overall habitat availability, the balance of native and non-native fish habitat, and potential for vegetation movement. Reducing high flows created more habitat for fish with non-natives having the largest increases and led to vegetation encroachment. Reducing baseflows did not affect vegetation habitat, but native and non-native fish habitat was greatly reduced.

## ACKNOWLEDGEMENTS

Funding for this project was provided by the U.S. Forest Service and the U.S. Geological Survey. Data was made available through the work of students and faculty at University of Arizona and Colorado State University and staff at the U.S. Forest Service and U.S. Geological Survey.

I am incredibly thankful for my advisor Ryan Morrison who gave me the opportunity to work on this project. Thank you for being a source of guidance and encouragement during my time at CSU. Your engagement, understanding, and flexibility helped me stay on track amid a pandemic and university shutdown. I would also like to thank Matthew Ross and Pierre Julien for lending their time and expertise by serving on my committee and Chris Holmquist-Johnson for working diligently to provide me with the data I needed to complete my project. I am grateful for the encouragement and support I received from my friends, family, and partner over the past two years. Thank you all for helping me keep an optimal work-life balance.

## TABLE OF CONTENTS

|   |     |
|---|-----|
| ABSTRACT.....                                     | ii  |
| ACKNOWLEDGEMENTS .....                            | iii |
| 1 Introduction.....                               | 1   |
| 2 Study area .....                                | 5   |
| 3 Methods.....                                    | 6   |
| 3.1 REF DSSV2 Framework .....                     | 6   |
| 3.2 Hydrodynamic Modeling.....                    | 8   |
| 3.3 Module 1: Pre-processing .....                | 9   |
| 3.4 Module 2: Instream Habitat Processing.....    | 9   |
| 3.5.1 Calculation of instream fish habitat .....  | 11  |
| 3.6 Module 3: Time Series Analysis .....          | 12  |
| 3.6.1 Fish Habitat Processing.....                | 14  |
| 3.6.2 Floodplain Vegetation Processing .....      | 14  |
| 3.7 Module 4: Scenario Comparison .....           | 16  |
| 4 Results.....                                    | 17  |
| 4.1 Baseline Fish Habitat .....                   | 17  |
| 4.2 Baseline Vegetation Habitat .....             | 20  |
| 4.3 Scenario Evaluation .....                     | 20  |
| 5 Discussion .....                                | 24  |
| 5.1 Limitations of REF DSSV2 .....                | 27  |
| 5.2 Habitat Restoration and Protection .....      | 28  |
| 6 Conclusions .....                               | 29  |
| References .....                                  | 31  |
| Appendix A: REF DSSV2 Basecode .....              | 35  |
| Appendix B: REF DSSV2 Function Descriptions ..... | 82  |
| Appendix C: REF DSSV2 Quick Start Tutorial .....  | 87  |

## **1 Introduction**

Commercial and residential development continues to put pressure on riverine ecosystems. Limited water resources are obligated in fulfillments for irrigation, municipal water supply, and other human needs, leaving little to be allocated for environmental uses (Richter and Orr, 2017). Increased variability in water supply (Lytle et al., 2017) and other predicted hydrologic changes such as diversion projects or reservoir operations also need be considered in management efforts (Poff et al., 2010). Environmental flow management seeks to mitigate the harm on these ecosystems caused by hydrologic alteration (Poff and Matthews, 2014).

Environmental flow studies and planning tools have largely focused on meeting the needs of high value or endangered fish (Rivaes et al., 2017). Hydraulic-habitat models are frequently used because they link physical characteristics of a stream to habitat suitability or availability (Lamouroux et al., 2017). These physical characteristics are often a function of discharge, which are linked through statistical analysis and numerical modeling. Statistical hydraulic-habitat models empirically link changes in characteristics like hydraulic geometry to changes in habitat suitability or availability (Lamouroux et al., 2017; Dunbar et al., 2012). Statistical approaches can incorporate expert knowledge and other qualitative information, so they are common where field data is limited (Dunbar et al., 2012; Marsili-Libelli et al., 2013).

Numerical 1D and 2D models require more topological and hydrological field data, but they provide better spatial representation of habitat (Lamouroux et al., 2017). A key output of these models is available habitat. Weighted useable area (WUA) is a common measure of available habitat. Habitat suitability indices ranging from 0 to 1 are multiplied by cross-sectional area (Milhous et al., 1984) or reach surface area (Yao et al., 2018) to calculate WUA. When habitat suitability criteria are delivered in binary format, meaning habitat quality is not considered (Bovee, 1986), habitat area may simply be the sum of suitable area in the reach (Bovee and Waddle, 2010). Habitat time series, first introduced in the Instream Flow Incremental Methodology (IFIM), is another metric of available habitat designed to compare the

effects of alternative flow scenarios (Milhous et al., 1984). The corresponding habitat area value for all discharges in a flow scenario is used to report habitat area at every time step (Bovee and Waddle, 2010). A combination of these statistical and numerical approaches is also used in recent instream habitat studies (Marsili-Libelli et al., 2013; Lamouroux et al., 2017; Yao et al., 2018).

The ecological needs of floodplain ecosystems are neglected when flows are managed solely for instream habitat needs (Tonkin et al., 2020). Floodplain vegetation habitats operate on longer lifecycles that depend on inter-annual flow variability (Rivaes et al., 2017; Lytle et al. 2017; Tonkin et al., 2020), and are especially sensitive to changes in large overbank flows (Morrison et al., 2014). Tonkin et al. (2020) suggested that managing for floodplain habitat also fulfills the needs of instream habitats making them a suitable environmental indicator.

There are a variety of methods for determining the environmental flow needs of floodplain vegetation. These methods are discussed at length in Merritt et al. (2010). Vegetation has been linked to the flow regime through numerical (Baird et al., 2005) and stochastic (Lytle and Merritt, 2004) modeling, regression analysis (Gonzalez et al., 2012), and Bayesian networks (Morrison and Stone, 2014). These predictive models are available from the individual to community levels (Merritt et al., 2010).

Methods providing spatially explicit information are valued for their ability to describe changes in the extent and distribution of vegetation (Merritt et al., 2010). Models linking vegetation to the flow regime can be combined with spatially grounded models to provide spatial results. For example, Morrison and Stone (2014) used a Bayesian network and 2D hydrodynamic model to make predictions about vegetation recruitment. In a decision support system designed for the Connecticut River Basin, hydraulic models were linked with floodplain plant community models to assess changes in areas occupied by those communities (Julian et al., 2016). The spatial components included in these tools made it possible for the spatial

dynamics of floodplain vegetation to be better predicted (Morrison and Stone, 2014; Julian et al., 2016).

There are limited environmental flow planning tools available that incorporate both instream and floodplain habitats. Optimization models such as those developed by Alafifi and Rosenberg (2020) and Steinschneider et al. (2014) seek to recommend flows that optimize ecological benefits while operating within the constraints of reservoir management. Some of these tools focus on maximizing ecological benefits, others strive to also maximize economical and recreational values (Alafifi and Rosenberg, 2020). Holistic scenario comparison tools also evaluate ecological targets in the context of reservoir management constraints. DRIFT, a decision support system designed for larger-scale environmental flow assessments, includes fish and vegetation modules in its holistic evaluation of river systems (Southern Waters, 2017). DRIFT has primarily been applied in basin level projects in developing countries (Southern Waters, 2017). Its outputs are suitable for tradeoff evaluation (Arthington et al., 2007) in a variety of project types. Outputs come in the form of qualitative descriptions of ecological change. Other scenario comparison tools focus on ecological impacts only. These flow scenarios account for management limitations. Ecosystem Functions Model (HEC-EFM) is a tool to help determine ecological response to flow alteration (Hickey, 2017). HEC-EFM combines qualitative and quantitative analyses through incorporation of hydraulic modeling, expert knowledge, and expected hydrologic alteration. Another notable feature of the tool is the ability to identify promising restoration sites. The Computer Aided Simulation Model for Instream Flow and Riparia (CASiMiR) software suite has fish and vegetation models available. Fish habitat requirements can be input as both qualitative and quantitative information. The vegetation model is most appropriate for temperate and alpine environments (Ecohydraulic Engineering, GmbH, 2019). Spatial analysis is fully integrated into the models. Outputs include maps and tables that make it easy for users to understand the modeling results. Vegetation and



fish habitat models are uncoupled. Analysis must be completed separately, and their results compared.

Existing tools have several limitations. In holistic evaluation tools, instream and floodplain habitat benefits may compete against economic or recreational gains. Additionally, outputs can be difficult to quantify, and resulting metrics may be difficult for stakeholders to interpret (Alafifi and Rosenberg, 2020). Alafifi and Rosenberg (2020), Steinschneider et al. (2014), DRIFT, and HEC-EFM also have limitations on spatial analysis. DRIFT, Alafifi and Rosenberg (2020), and Steinschneider et al. (2014) do not have a direct spatial links, preventing habitat from being mapped and changes from being tracked spatially. HEC-EFM includes a spatial module that allows users to complete these tasks. However, this module is not fully integrated. In order to access their spatial module HEC-GeoEFM, users must have access to ArcGIS.

My thesis describes the development and application of the Riverine Environmental Flow Decision Support System, version 2.0 (REFDSSV2), which is an open-source tool for evaluating the trade-offs of fish and vegetation habitat in alternative flow scenarios. The objectives of this tool are to 1) develop an open-source decision support tool that can be used to evaluate fine-scale impacts of flow alterations on both instream and floodplain habitat, and 2) describe instream and floodplain habitat changes through simplified, relevant metrics. To demonstrate the functions of the REFDSSV2, I show results from a case study on the Verde Wild and Scenic River in Arizona, USA.

The REFDSSV2 builds off of REFDSSV1. REFDSSV1 was designed to evaluate the impacts of alternative flow scenarios on target organisms in the Delaware River (Maloney et al., 2015). Hydraulic habitat suitability criteria were linked with 2D hydrodynamic models using River2D to generated maps and metrics of available habitat (Maloney et al., 2015). Its strengths included representation of multiple life stages, habitat time series, and spatial representation of results. Analysis in the REFDSSV1 was limited to two alternative flow scenarios and instream

aquatic organisms only. The REFDDSSV2 seeks to include the strengths of REFDDSSV1 while expanding its capabilities, including the evaluation of floodplain vegetation habitat.

## 2 Study area

The Verde River is a perennial river that starts in Arizona's Central Highlands and flows 309 km south into the Salt River (Paretti et al., 2018). The typical flow regime includes high flow pulses from storms in the winter and late summer and sustained snowmelt fed high flows during the spring. Streamflow is maintained year-round by groundwater contributions (Paretti et al., 2018).

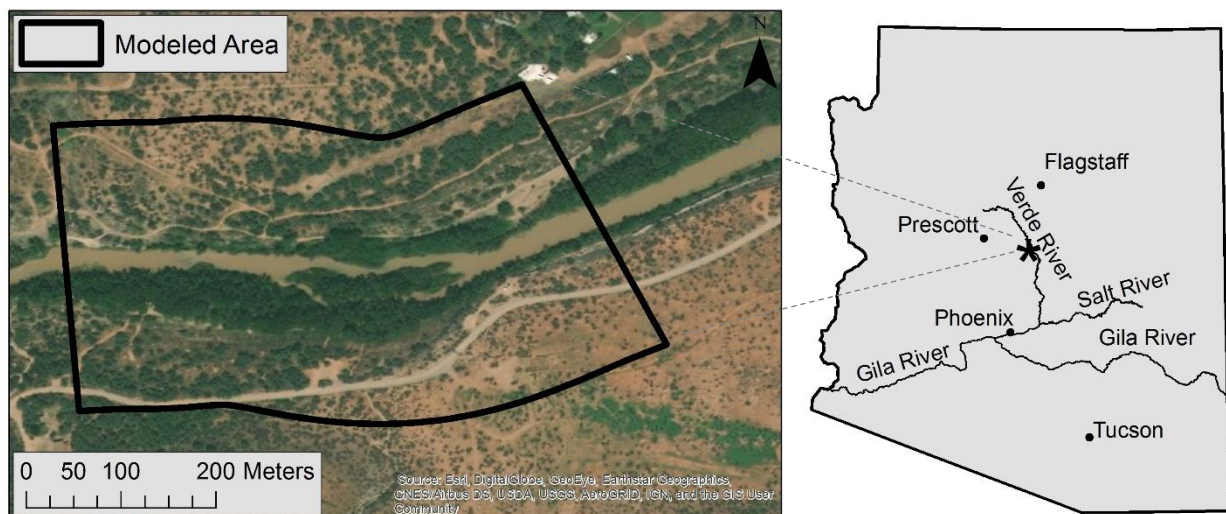


Figure 1. Overview of study area. The study reach is located at the beginning of the Wild and Scenic stretch of the river.

The Verde River corridor has high recreational, ecological, and cultural value. It supports a diverse collection of fish, wildlife, and vegetation including over 50 special status fish and wildlife species (U.S. Forest Service, 2004). Additionally, it contributes critical drinking and irrigation water to the Salt River Project. The Verde River is under pressure from development and population growth in surrounding areas (Paretti et al., 2018).

The Wild and Scenic designation applies to a 65.2 km stretch in the lower portion of the river with 35.7 km classified as Wild and 29.5 km classified as Scenic (U.S. Forest Service, 2004). This designation is used to preserve rivers in “free-flowing condition... for the benefit and enjoyment of present and future generations” (Wild & Scenic Rivers Act, Oct 2, 1968). My study reach is located near Beasley Flat, which marks the beginning of the Wild and Scenic stretch of the river (Figure 1). The reach is approximately 1 km long. The study area includes the relatively narrow adjacent riparian zone and extends into the floodplain, and the channel includes a narrow, vegetated island. The riparian zone is populated by numerous tree species, including willows, velvet ash, Fremont cottonwood and Arizona sycamore, as well as herbaceous vegetation, including sedges, rushes, horsetail, cattail, and reeds (U.S. Forest Service, 2004).

Data used to describe this site was made available through a series of field visits and continuous monitoring. USGS gage #09506000 (Verde River near Camp Verde, AZ) provided the historic flow record at the site. Median daily discharge was available beginning in October 1988; I included flow records through June 6, 2019. The median annual flow at this site is 4.5 m<sup>3</sup>/s. Flows typically peak in March and decline through July, rising again in the fall. River transducers installed by USGS and USFS collaborators near the upstream and downstream ends of the study reach collected stage information used to develop a site-specific stage-discharge relationship. Groundwater wells installed by USFS and USFS collaborators were used to model surface water-groundwater interactions. Topographic data was collected by ground-surveys and UAS imagery.

### **3 Methods**

#### **3.1 REFDDSSV2 Framework**

An overview of the major inputs and outputs to the REFDDSSV2 is shown in Figure 2. The REFDDSSV2 was split into four modules: pre-processing, instream habitat processing, time series analysis, and scenario comparison. Each module is dependent on the results of the preceding module. The main inputs to the REFDDSSV2 are a series of 2D hydrodynamic

modeling results, habitat preference information for fish and vegetation, the historic flow record, and alternative flow scenarios. I describe each of these components in Table 1, including the source of data from the Beasley Flat study reach.

Table 1. Required data inputs for REFDSV2.

| <b>Data Name</b>                               | <b>Description</b>   | <b>Source</b>   |
|--|--|---|
| Stage-discharge curves                         | Used for calibration of 2D hydrodynamic models   | USGS gage #09506000; water level loggers installed by collaborators   |
| Site topography                                | High resolution coverage spanning channel, riparian zone, and floodplain   | RTK ground survey; structure-from-motion surface using Unmanned Aerial Systems imagery collected by USGS collaborator (Chris Holmquist-Johnson) |
| 2D hydrodynamic modeling results               | 0.3 m by 0.3 m delimited results for discharges spanning historic conditions   | FaSTMECH modeling results produced by USGS collaborator (Chris Holmquist-Johnson)   |
| Fish habitat suitability criteria              | Binary habitat preference models for fish residing in study reach  | Fish surveys collected by University of Arizona collaborator (Scott Bonar); previous studies  |
| Vegetation probability of occurrence functions | Logistic regression equations relating probability of occurrence of vegetation to the exceedance probability of daily habitat inundation | Plant surveys collected by Colorado State University collaborator (Erin Cubley); exceedance probability curves                                  |
| Historic flow record                           | Time series of daily discharge spanning several years  | USGS gage #09506000   |
| Flow scenarios                                 | Multi-year time series of daily discharge simulating alternative management or climate scenarios   | Stakeholders  |

These inputs required collaboration with fish ecologists, riparian ecologists, and hydraulic engineers from the USGS, USFS, University of Arizona, and other CSU researchers. The quality of the 2D hydrodynamic modeling results is dependent on both data availability and the modeler's experience. A hydraulic engineer is best qualified to make judgements on modeling extent, resolution, and calibration. Fish and riparian ecologists should supply habitat preference information. They should also be consulted if preference information is being transferred between study areas. Initial results should be reviewed by ecologists to ensure that habitat quantity and location are reasonable.

### 3.2 Hydrodynamic Modeling

A two-dimensional hydrodynamic model was developed using the FaSTMECH modeling software by Chris Holmquist-Johnson (USGS) with my assistance. FaSTMECH is a quasi-steady morphological solver capable of simulating river flow and long-term morphological changes. The hydrodynamic portion of the solver uses the shallow-water equations in 2D (Nelson et al., 2016). The computational curvilinear grid had an approximate cell size of 0.25 by 0.25 meters and length of 1,000 meters. The model was run for a range of discharges spanning 0.5 m<sup>3</sup>/s to 1,500 m<sup>3</sup>/s which simulated the historic and potential conditions in the reach. The results of every run were output on the same computational grid and exported as Comma Separated Value (CSV) files.

We assumed there was no sediment transport in the reach, so only the hydrodynamic component of FaSTMECH was utilized. Survey data was supplemented with raw Unmanned Aerial Systems (UAS) data to generate a point file of the topography of the study reach in ArcGIS, and the point file was converted to a triangulated irregular network (TIN) surface. Break lines and additional nodes were added as needed. The final TIN was converted to a raster (0.3 m resolution) using the natural neighbor algorithm. The raster was converted to a point file that was brought into FaSTMECH.

A computational curvilinear grid with an approximate cell size of 0.25 m by 0.25 m was generated for the reach. Topographic information and initial roughness patches were mapped to the grid. Using the stage-discharge data, the model was calibrated by adjusting roughness values to minimize the root mean square errors between simulated and observed water surface elevations. The model was run using a python script at 125 different discharges spanning the entire range of the gage record.

### 3.3 Module 1: Pre-processing

The first module in the REFDSSV2 is the *pre-processing module* (Figure 2). This module processes the hydrodynamic modeling results and rasterizes substrate maps, if applicable. Habitat metrics, the variables included in habitat preference information, are rasterized using the Raster package in R (Hijmans et al., 2019). Depth and depth-averaged velocity results from the hydrodynamic model were also rasterized. A list of the modeled discharges is compiled from the names of the modeling result CSVs. The rasters are bilinearly interpolated to ensure complete coverage over the calculation grid. The REFDSSV2 will also rasterize a shapefile of substrate types, if it is provided. A map giving substrate type according to Wentworth grain size classification was provided and rasterized for the Verde case-study site. All the rasterized files are of the same coordinate reference system, extent, and resolution.

### 3.4 Module 2: Instream Habitat Processing

The second module in the REFDSSV2 is the *instream habitat processing module* (Figure 2). The purpose of this module is to combine habitat suitability criteria and the rasterized 2D hydrodynamic modeling results to calculate habitat availability for each species and life stage at every modeled discharge. Due to limited information about juvenile and larval stage fish, the Verde River case study is limited to analyses of adult fish.

A team led by Scott Bonar (University of Arizona) conducted fish surveys in 2017 and 2019 on the Verde River. Based on fish surveys and discussions with the collaborative team, HSCs were developed for the native and non-native species shown in Table 2. Suitable ranges

of velocity, depth, and substrate size were included in the HSC for each species. If sample sizes were insufficient, HSC was derived from previous studies in the Gila River Basin.

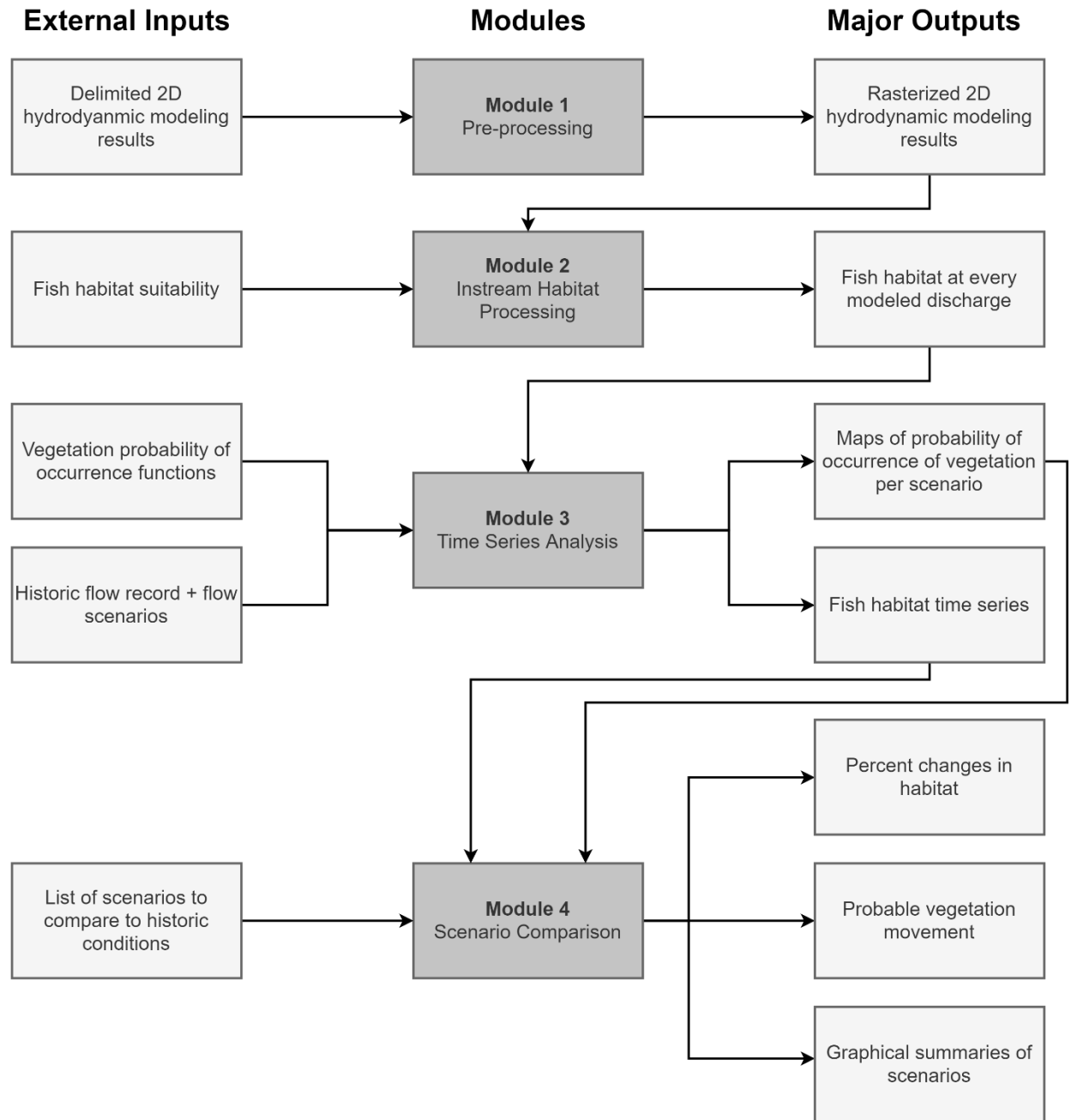


Figure 2. An overview of the major inputs and outputs to the four modules of the DSS. External inputs are inputs that need to be prepared outside the REFDSV2 environment.

Table 2. Habitat suitability requirements for fish. Depths beyond 2 meters were not measurable in the field, so Sonora sucker and Desert sucker have infinite maximum suitable depths.

| Species   | Status     | Velocity<br>(m/s) | Water<br>Depth<br>(m) | Substrate<br>Size<br>(mm) | Data Source(s)  |
|---|------------|-------------------|-----------------------|---------------------------|---|
| Desert<br>Sucker<br><br>( <i>Catostomus<br/>clarkii</i> )                         | Native     | 0.091 - 0.50      | 0.24 - 2+             | 2 - 256                   | Rinne, 1992; Paroz<br>et al., 2006; Bonar et<br>al, 2010; Nemec,<br>2018  |
| Roundtail<br>Chub<br><br>( <i>Gila<br/>robusta</i> )                              | Native     | 0 - 0.22          | 0.41 - 2+             | 0.06 - 256                | Bestgen and Propst,<br>1989; Rinne, 1992;<br>Barret and Maughan,<br>1995; Brouder, 2000;<br>Paroz, 2006; Bonar<br>et al., 2010; Jenney,<br>2019 |
| Speckled<br>Dace<br><br>( <i>Rhinichthys<br/>osculus</i> )                        | Native     | 0.084 - 0.44      | 0.18 - 0.41           | 0.06 - 256                | Rinne, 1992; Paroz<br>et al., 2006; Bonar et<br>al, 2010; Nemec,<br>2018; Jenney, 2019  |
| Sonora<br>Sucker<br><br>( <i>Catostomus<br/>insignis</i> )                        | Native     | 0.029 - 0.45      | 0.23 - 2+             | 0.06 - 256                | Rinne, 1992; Paroz<br>et al., 2006; Bonar et<br>al, 2010; Nemec,<br>2018; Jenney, 2019  |
| Red Shiner<br><br>( <i>Cyprinella<br/>lutrensis</i> )                             | Non-native | 0.13 - 0.51       | 0.29 - 0.43           | topsoil - 64              | 2019 and 2017 field<br>data   |
| Smallmouth-<br>Redeye Bass<br><br>( <i>Micropterus<br/>dolomieu-<br/>coosae</i> ) | Non-native | 0.045 - 0.28      | 0.32 - 0.54           | 0.06 - 256                | 2019 and 2017 field<br>data   |

### 3.5.1 Calculation of instream fish habitat

Suitable habitat is determined by cross-referencing the HSC with the rasterized 2D hydrodynamic modeling results and a map of substrate types in the reach. A series of raster



operations are used to identify suitable areas on a cell-by-cell basis for every discharge. These processes are repeated for every species and life stage. The resulting rasters match the resolution and extent of the rasterized modeling results. Cells that do not completely meet the suitability requirements are left empty.

Next, isolated cells or habitat patches, hereon referred to as islands, were identified. Islands are patches of habitat that are not adjacent to any other patches of habitat. Since these patches may be inaccessible from other habitat patches, islands that are too small to sustain fish are removed to avoid an overestimation of available habitat. The island size threshold is specified by the user and should be based on the raster resolution and species in consideration. On the Verde River case study, islands consisting of two cells or less ( $< 0.5 \text{ m}^2$ ) were removed because the collaborative team did not think that area was sufficient to support adults of the larger species we considered. For example, the Sonora Sucker's maximum body length exceeds the length of two cells (U.S. Fish and Wildlife Service, 2009).

Habitat area is tabulated by multiplying the number of populated cells by the resolution of the raster for each layer. Habitat area-lookup tables are developed for each species and life stage. These tables contain length-normalized habitat area for each modeled discharge. This area is calculated by dividing total available area by reach length. Normalizing habitat area enables comparison between multiple reaches or river systems. Habitat area-discharge curves are plotted from the habitat area-lookup tables.

### 3.6 Module 3: Time Series Analysis

Module 3 of the REFDDSV2 is the *time series analysis module* (Figure 2). The purpose of this module is to determine available habitat for each scenario. Alternative flow scenarios are time series of daily discharge that span at minimum several years. The historic flow record (1989 to 2019) served as the baseline scenario. User-supplied scenarios may simulate the effects of climate change, reservoir operations, withdrawals, and more. I considered two

alternative flow scenarios described in Table 3. The differences between these scenarios are shown in Figure 3.

Table 3. Descriptions of flow scenarios compared to historic baseline conditions from USGS gage #09506000. Alternative flow scenarios were derived from the historic flow record from April 1, 1989 to June 1, 2019. In Scenario A high flows were not reduced below the median discharge. In Scenario B baseflows were flows below the 25<sup>th</sup> percentile of the historic flows.

| Scenario   | Purpose  | Description   |
|------------|--|---|
| Baseline   | Evaluate historically available habitat area and determine observed flow metrics | Baseline conditions observed over the past 30 years                         |
| Scenario A | Evaluate effects of reductions in high flows                                     | 25% reduction in flows above the observed median discharge.                 |
| Scenario B | Evaluate effects of reductions in baseflows                                      | 20% reduction in flows below the 25th percentile of historic observed flows |

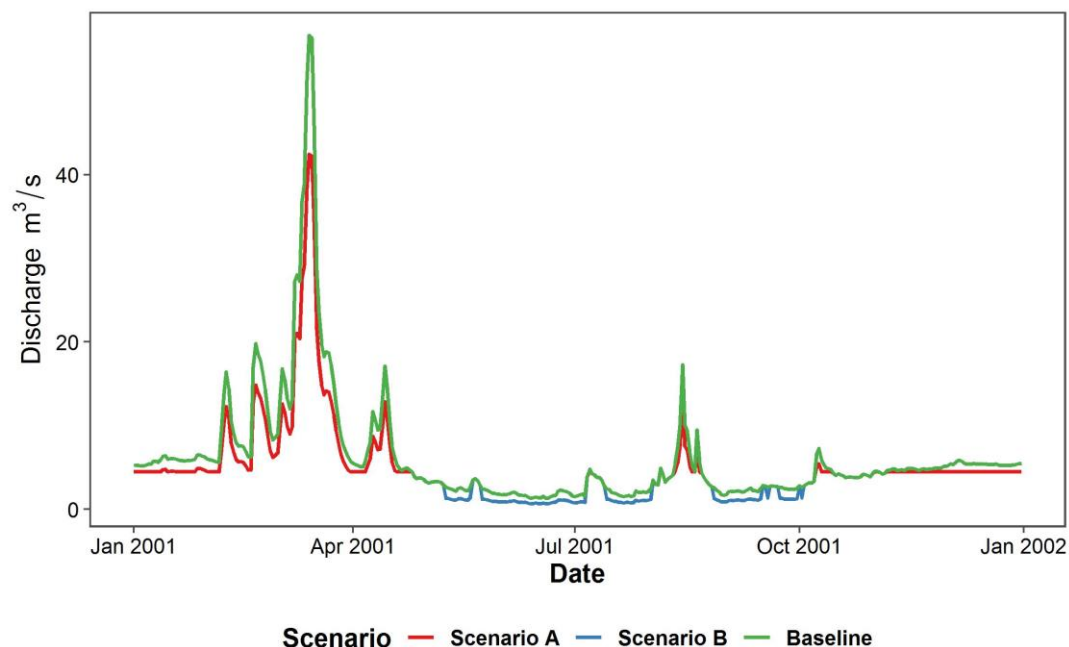


Figure 3. Flow scenarios in 2001. Scenario A reduced high flows and Scenario B reduced base flows.

### 3.6.1 Fish Habitat Processing

Changes in fish habitat for each scenario are analyzed using two approaches. First, the cumulative change in fish habitat is quantified through the entire period of record for each scenario. Flow scenarios are provided as time series of daily discharge. Every unique discharge from the hydrograph and habitat area-lookup table is put into a single list and arranged by increasing discharge. An entry of zero discharge with zero available habitat area is added to the table, to address discharges below the lowest modeled discharge. Linear interpolation is used to calculate a length normalized habitat area value for every unique discharge in the hydrograph using values from the habitat area-lookup table. Daily habitat area is added to the hydrograph by matching discharge values to those in the table of interpolated areas.

Second, the minimum 10-day running mean habitat area value is calculated for each month. This value, referred to as min10, represents the limiting period in a month (Waddle and Bovee, 2010). The central running mean is calculated for every day in the flow scenario as the average of the four previous days, the current date, and the five following days.

### 3.6.2 Floodplain Vegetation Processing

Vegetation response was predicted for guilds and individual species. Vegetation data was collected by U.S. Forest Service personnel by plots along transects stretching from the edge of the channel to the upland. Plants were grouped by their traits into functional guilds (Merritt et al., 2010). Traits included their morphology and hydrologic and disturbance tolerances. I analyzed two guilds: Tall trees and drought tolerant shrubs. Three key species in these guilds were also analyzed. Velvet mesquite (*Prosopis velutina*) belonged to the drought tolerant shrubs guild, and Fremont's cottonwood (*Populus fremontii*) and Goodding's willow (*Salix gooddingii*) belonged to the tall trees guild.

Probability of occurrence of these functional guilds and species was related to daily exceedance probability (EP) of habitat inundation (Diehl et al., 2017) using logistic regression equations listed in Table 3. These equations were derived in R using the caret package (Kuhn,

2019) by CSU collaborator, Erin Cubley. Plant presence or absence information was used to verify the accuracy of the models. Daily exceedance probability ( $EP$ ) was chosen as the explanatory variable because plant communities' spatial distribution often follows gradients of flood frequency (De Jager et al., 2015).

Table 3. Logistic regression equations for the groups. Drought tolerant shrubs and tall trees are functional guilds.

| Group                   | Equation (Eq. 1-5)  |
|-------------------------|---|
| Drought Tolerant Shrubs | $\frac{\exp(-0.486-25.9EP)}{1+\exp(-0.486-25.9EP)}$               |
| Velvet mesquite         | $\frac{\exp(-0.653-31.7EP)}{1+\exp(-0.653-31.7EP)}$               |
| Tall Trees              | $\frac{\exp(-2.57+379EP-4198EP^2)}{1+\exp(-2.57+379EP-4198EP^2)}$ |
| Fremont's cottonwood    | $\frac{\exp(-2.23+141EP-1517EP^2)}{1+\exp(-2.23+141EP-1517EP^2)}$ |
| Goodding's willow       | $\frac{\exp(-2.66+226EP-2604EP^2)}{1+\exp(-2.66+226EP-2604EP^2)}$ |

Daily  $EP$  was calculated using Weibull plotting position (Eq. 6) for every discharge in each flow scenario where  $m$  was the rank of the discharge and  $N$  was the number of discharges in the flow scenario (Cunnane, 1978).

$$EP = \frac{m}{N+1} \quad \text{Eq. 6}$$

Next, daily  $EP$  was calculated for the modeled discharges in the context of each scenario.  $EP$  for the modeled discharges was necessary to spatially link flow scenarios, vegetation, and the study area since inundation information was only available for modeled discharges. For modeled discharges within the range of the scenario, linear interpolation was used.

An *EP* of 1, meaning the cells were always inundated, was given to discharges below the range of the scenario. *EP* was calculated for discharges above the range assuming one event per discharge. Since Weibull plotting position is a function of frequency rather than magnitude, the same discharge could have a different *EP* in different scenarios.

A single map of inundating discharge was generated for the reach. The value in each cell in this raster was the modeled discharge that first made a cell's depth greater than zero. This layer will be referred to as the Digital Inundation Map (DIM). The DIM was reclassified using *EP* of the modeled discharges. The resulting layer gave the daily *EP* of habitat inundation at every cell.

Maps of probability of occurrence of each vegetation group were generated from the maps of *EP*. Probability of occurrence was calculated for every modeled discharge using the Eq. 1–5 in Table 3. The *EP* maps were reclassified using these probabilities giving maps of probability of occurrence of vegetation groups residing in a cell.

### 3.7 Module 4: Scenario Comparison

Scenarios were evaluated on their performance in three categories: 1) percent change in habitat area, 2) balance of native and nonnative fish habitat, and 3) potential vegetation habitat movement. These categories were represented through a combination of numerical and visual indicators. The relative importance of each of these categories may be determined by the stakeholders.

Percent change in habitat area from baseline conditions was calculated for both fish and vegetation. The difference between the scenario provided habitat and the baseline habitat was divided by the baseline habitat.

For fish, min10 was the habitat value used. Percent change was calculated for each month in the scenario. These months were averaged for all years in the scenario to get an average percent change in min10 in a scenario. An overall average percent change value was also calculated. Fish species were designated as native or non-native.

Areas with a probability of occurrence of a group at 0.70 or higher were considered suitable vegetation habitat (Personal communication with David Merritt, USFS, March 2020). A single habitat area value was tabulated for each group for every scenario and used to calculate percent change in habitat area.

Vegetation movement was determined using the distribution of inundating discharges found in areas of high probability of occurrence. Inundating discharge represented position on the floodplain where higher inundating discharges were associated with greater distances from the channel. The maps of suitable vegetation habitat were laid over the DIM to get a distribution of inundating discharges for each group for every scenario. The first and third quartiles of these distributions were compared in an effort to capture changes in the center of the distribution. Vegetation habitat was described as expanding, encroaching, retreating, reducing, or not changing based on this comparison (Table 4).

Table 4. Criteria for probable vegetation movement descriptions. Movement was based on changes in the distribution of discharges that inundated their most probable habitat.

| <b>Description</b> | <b>Change in 1st quartile</b> | <b>Change in 3rd quartile</b> |
|--------------------|-------------------------------|-------------------------------|
| Expand             | decrease                      | increase                      |
| Encroach           | decrease                      | decrease                      |
| Retreat            | increase                      | increase                      |
| Reduce             | increase                      | decrease                      |
| No change          | no change                     | no change                     |

## **4 Results**

### **4.1 Baseline Fish Habitat**

Available habitat varied with discharge for every species I considered (Figure 4). There were several optimal ranges of discharge for each species. All species had a relatively high amount of habitat available at flows below the median annual discharge of 4.5 m<sup>3</sup>/s. Several

species had their minimum or near minimum habitat availability at 0.5 m<sup>3</sup>/s, the lowest modeled discharge. Red Shiner and Desert Sucker minimum availabilities at this discharge were 590 m<sup>2</sup>/km and 1,737 m<sup>2</sup>/s. Beyond baseflow (0.5 to 2.5 m<sup>3</sup>/s) Sonora Sucker and Desert Sucker consistently had greater habitat availability than any other species. Their hydraulic habitat preferences had the widest ranges of depths and velocities of the species considered. Roundtail Chub, which prefer slower moving waters, had its maximum habitat availability of 15,453 m<sup>2</sup>/km at 3 m<sup>3</sup>/s.

Habitat gradually dropped off after overbank flow began at 8.5 m<sup>3</sup>/s. Available habitat hit local minimums at 35 m<sup>3</sup>/s. Discharges between 50 and 120 m<sup>3</sup>/s produced optimal conditions for Speckled Dace, Smallmouth-Redeye Bass, Red Shiner, and Roundtail Chub. Smallmouth-Redeye Bass's maximum habitat availability was 7,183 at 65 m<sup>3</sup>/s. Desert Sucker and Sonora Sucker had a similar peak between 70 and 155 m<sup>3</sup>/s. Desert Sucker and Sonora Sucker had their maximum amounts of habitat available at 120 m<sup>3</sup>/s with 34,163 m<sup>2</sup>/km and 31,523 m<sup>2</sup>/km available. There was relatively low habitat availability between 200 m<sup>3</sup>/s and 600 m<sup>3</sup>/s for all species. Sonora Sucker and Speckled Dace had their lowest availability at 600 m<sup>3</sup>/s with 8,097 m<sup>2</sup>/km and 1,526 m<sup>2</sup>/km available.

Habitat availability increased again with discharge as new habitat was opened on the floodplain. Habitat availability was similar in magnitude for flows between 675 and 775 m<sup>3</sup>/s and 50 and 120 m<sup>3</sup>/s for all species. Discharges up to 1,000 m<sup>3</sup>/s maintained optimal conditions for Desert Sucker and Sonora Sucker. Roundtail Chub and Smallmouth-Redeye Bass, fish that prefer slower moving water, had minimal habitat availability for the remaining modeled discharges. Smallmouth-Redeye Bass had its minimum area at 1,000 m<sup>3</sup>/s with 550 m<sup>2</sup>/km available, the smallest habitat area of any fish, while Roundtail Chub had minimum availability at 1,500 m<sup>3</sup>/s with only 673 m<sup>2</sup>/km.

Speckled Dace and Red Shiner had variable habitat availability for the remaining modeled discharges. Both species had their highest habitat availability at 1,100 m<sup>3</sup>/s; Speckled

Dace had 10,359 m<sup>2</sup>/km and Red Shiner had 7,215 m<sup>2</sup>/km. Although these were their maximum values, they were similar in magnitude to peaks at 75 m<sup>3</sup>/s and 700 m<sup>3</sup>/s.

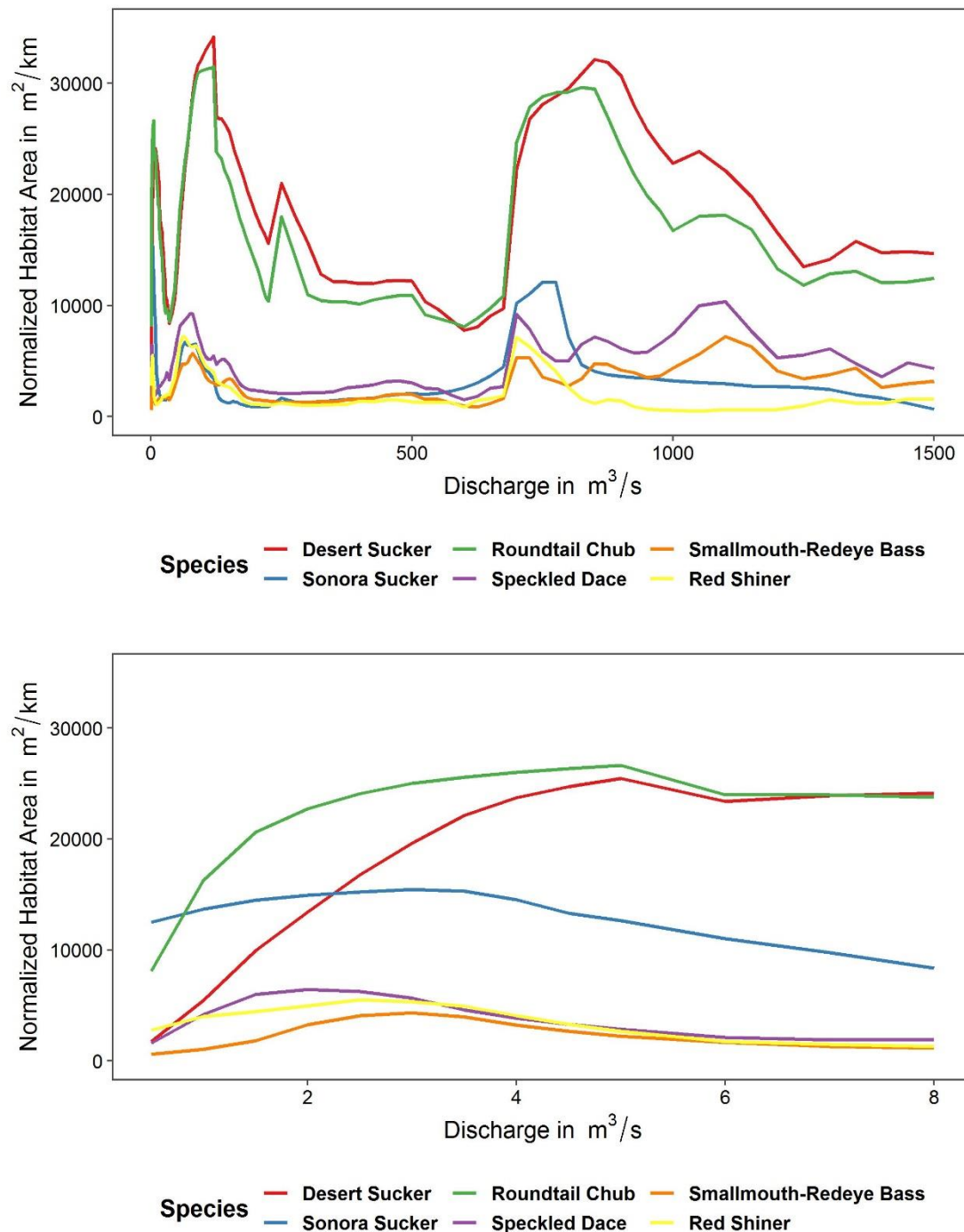


Figure 4. Habitat-area discharge curves for all species. The top panel shows all modeled discharges, and the bottom panel shows in-channel modeled discharges. Smallmouth-Redeye bass and Red Shiner are non-native.



## 4.2 Baseline Vegetation Habitat

Maps of probability of occurrence made from the historic flow record characterized the baseline vegetation habitat distribution. Drought tolerant shrubs, which includes *Prosopis velutina*, did not have probability of occurrences exceeding 0.38 based on historic conditions. *Prosopis velutina*'s maximum probability of occurrence was 0.34. The highest probabilities of occurrence were in areas furthest away from the channel where inundation was least likely to occur. Therefore, drought tolerant shrubs were not expected to be abundant in the study reach. Tall trees, including Goodding's willow and Fremont's cottonwood, all had areas of high probability of occurrence. Tall trees had the highest probability area (38,897 m<sup>2</sup>/km) followed by Goodding's willow (23,686 m<sup>2</sup>/km) and Fremont's cottonwood (7,013 m<sup>2</sup>/km). Tall trees were most likely to occupy areas inundated by discharges ranging from 19 to 105 m<sup>3</sup>/s which translates to areas inundated 8% to 1% of the time. Goodding's willow and Fremont's cottonwood were more likely in areas that were more frequently inundated. Goodding's willow's high probability areas were inundated by 23 to 60 m<sup>3</sup>/s of discharge and Fremont's cottonwood's high probability areas were inundated by 27 to 40 m<sup>3</sup>/s of discharge.

## 4.3 Scenario Evaluation

Scenario A led to an increase or no change in habitat availability for the fish I considered. Overall, habitat increased more in the fall (September - November) and winter (December – February) than in the spring (March – May) and summer (June – August) (Figure 6). Habitat in May through July when flows are typically at their lowest was the least impacted for these species.

The two non-native species Smallmouth-Redeye Bass and Red Shiner had the greatest magnitudes of change from this scenario. Smallmouth-Redeye Bass, which preferred slower waters, had habitat increase by over 30% November through February when high flows were reduced. Red Shiner habitat increased by over 20% during the same period.

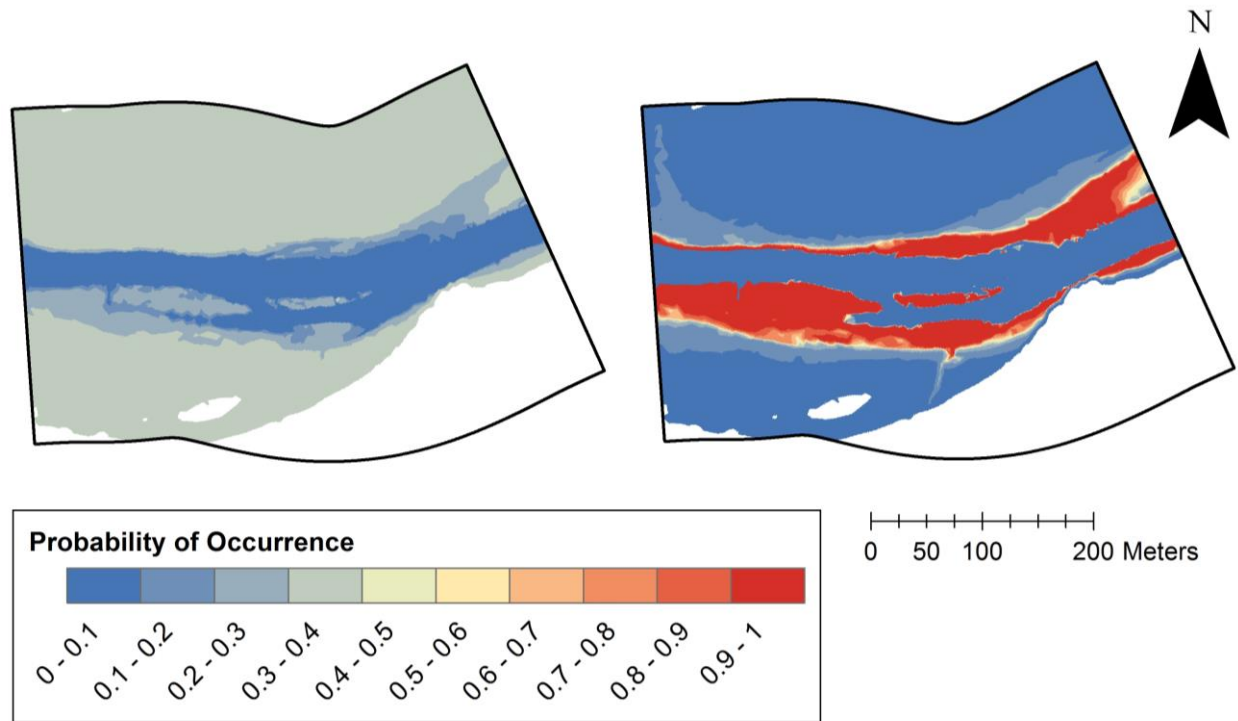


Figure 5. Maps of probability of occurrence for drought tolerant shrubs (left) and tall trees (right) for the baseline scenario. Drought tolerant shrubs did not have any areas considered high probability of occurrence.

Speckled Dace and Roundtail Chub also had relatively high increases in habitat November through February. Although Speckled Dace had habitat gains ranging from 15 to 33% during this period, between March and October it did not have any habitat gains of 10% or more. This period was dominated by lower flows. Roundtail Chub and Smallmouth-Redeye Bass habitat increased by more than 10% for 8 months out of the year. Neither Sonora Sucker nor Desert Sucker had an increase in habitat of 10% or more during any month with Scenario A (Figure 7).

Scenario B led to decreases in habitat availability for fish. Habitat losses were concentrated in April through October when baseflows dominated the hydrograph. Habitat loss was not predicted for any species in the period between November and February.

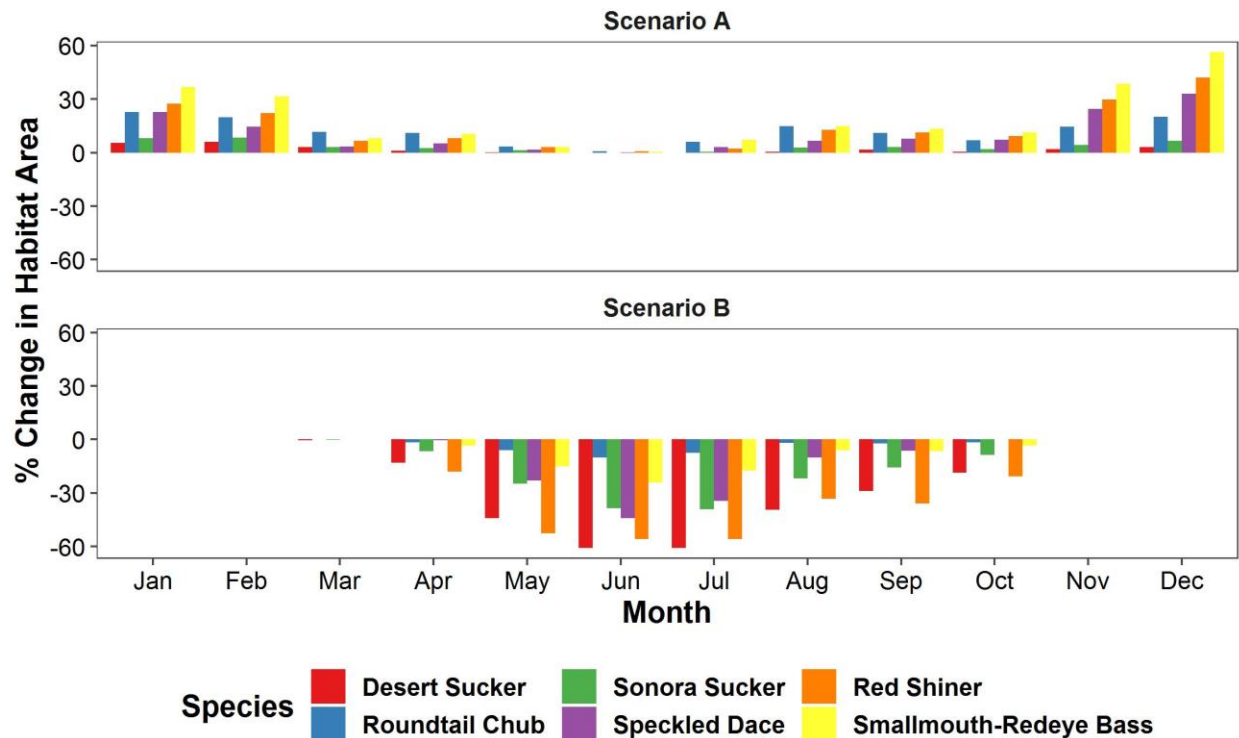


Figure 6. Percent change in normalized habitat area by month for all species. Native species are shown at the front of each month. Missing bars indicate 0% change.

Desert Sucker and Sonora Sucker lost large portions of their habitat under Scenario B. Desert Sucker lost 60% of their habitat in June and July and over 25% of their habitat in May, August, and September. Sonora Sucker lost over 20% of its habitat May through August with June and July habitat decreasing by more than a third. Speckled Dace also lost over 20% of its habitat in May through July, but April and October habitat availability was unaffected. Roundtail Chub had the least habitat loss for a native species; June was the only month where habitat loss was at least 10%.

Red Shiner habitat was reduced by over half May through July, a third in August and September, and nearly a fifth in April and October. Smallmouth-Redeye Bass had relatively small habitat losses. Only May through June had losses exceeding 10%. Similar to the other species, its greatest habitat loss was in June (24%).

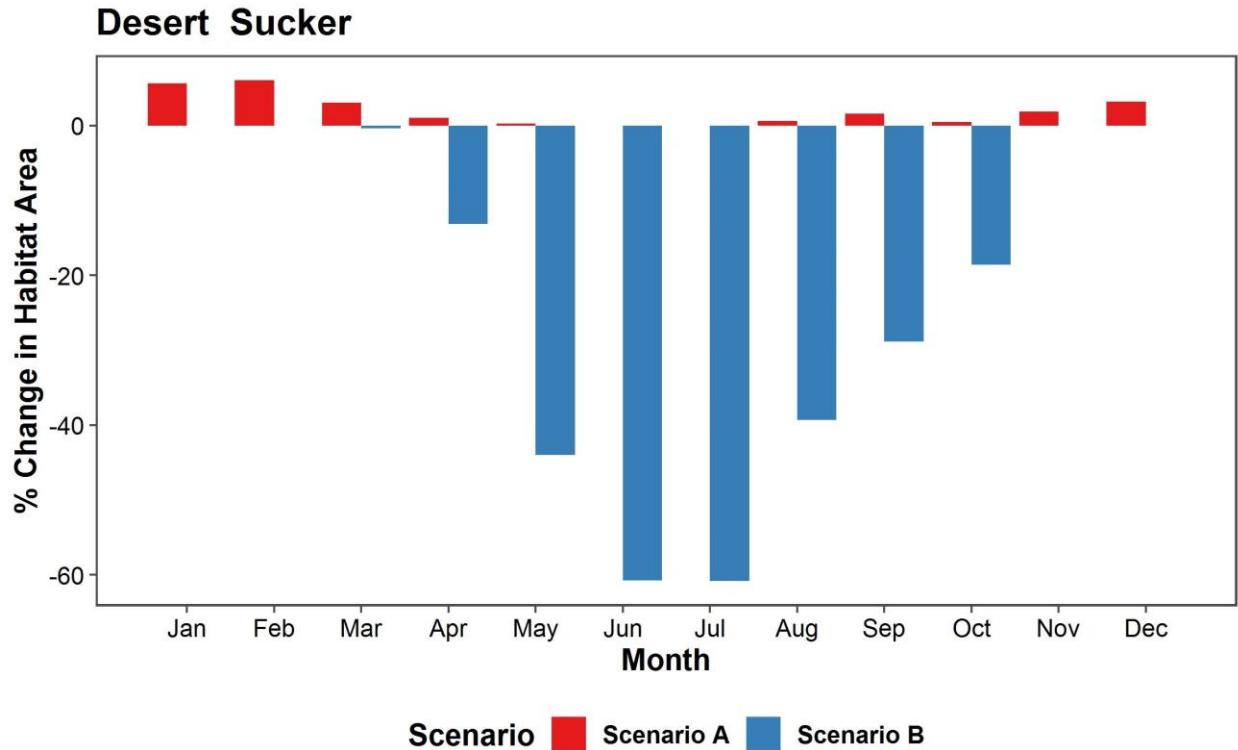


Figure 7. Percent change in normalized habitat area by month for adult Desert Sucker. Desert Sucker is native to the study area. Missing bars indicate 0% change.

Vegetation was more likely to be found in areas closer to the channel than it historically had with under Scenario A, likely because this scenario decreased *EP* for flows above the median discharge. For example, 5 m<sup>3</sup>/s went from a baseline *EP* of 0.43 to 0.20. This effect was less pronounced as discharges increased. Therefore, areas inundated by flows above the median discharge had the most change in probability of occurrence.

Probability of occurrence increased along the channel and decreased on the floodplain for species belonging to the tall trees guild. The tall trees guild had an overall 8% decrease in available high probability area. Goodding's willow high probability area decreased by more than a third. Fremont's cottonwood had a 7% increase in high probability area with most of the new area adjacent to the channel.

Species belonging to the drought tolerant shrubs guild did not have any areas with high probability of occurrence in Scenario A. Although the maximum probability of occurrence did not

change, probabilities did increase by up to 5% in areas inundated by flows between median and bankfull discharge.

Habitat availability and location of the vegetation I analyzed did not change under Scenario B. Scenario B primarily decreased *EP* values for baseflows. Since the recurrence of high flows was unaffected, vegetation did not respond to decreases in baseflow.

Generally, the decreased high flows in Scenario A led to an increase of fish habitat and encroachment of vegetation (Figure 8). Non-native fish, which were less tolerant of higher flows and depths, gained the most habitat in Scenario A. Overall gains by native fish were relatively small. Fremont's cottonwood also had relatively small gains. Conversely, the tall trees guild and Goodding's willow lost habitat. Movement into historically unoccupied areas accompanied changes in vegetation habitat.

While reduced baseflows in Scenario B did not affect vegetation, fish habitat availability declined. The overall losses for fish shown in Figure 8 included winter months with no habitat loss. Summer losses reached 60% for Desert Sucker and 40% for Speckled Dace; both species are native. While non-native species Red Shiner and Smallmouth-Redeye Bass had considerable losses as well, these declines came at the cost of sacrificing native fish habitat.

## **5 Discussion**

The REFDDSSV2 is unique in its fine-scale assessment of both fish and floodplain habitat. This open-source tool uses high resolution 2D hydrodynamic modeling to deliver spatially explicit, quantifiable results. This modeling approach enables the REFDDSSV2 to represent microhabitat features in a longitudinally continuous form rather than by cross-section (Milhous et al., 1984; Marsili-Libelli et al., 2013). This high-resolution spatial representation helps highlight the differences between alternative scenarios (Morrison and Stone, 2014) particularly for floodplain vegetation. Metrics of fish and floodplain vegetation are quantified directly from these spatial results. A challenge of similar tools is communicating complicated or empirically-based metrics (Alafifi and Rosenberg, 2020). The REFDDSSV2's metrics are designed to be open-

ended and easy to communicate thus facilitating tradeoff evaluation in a variety of environments.

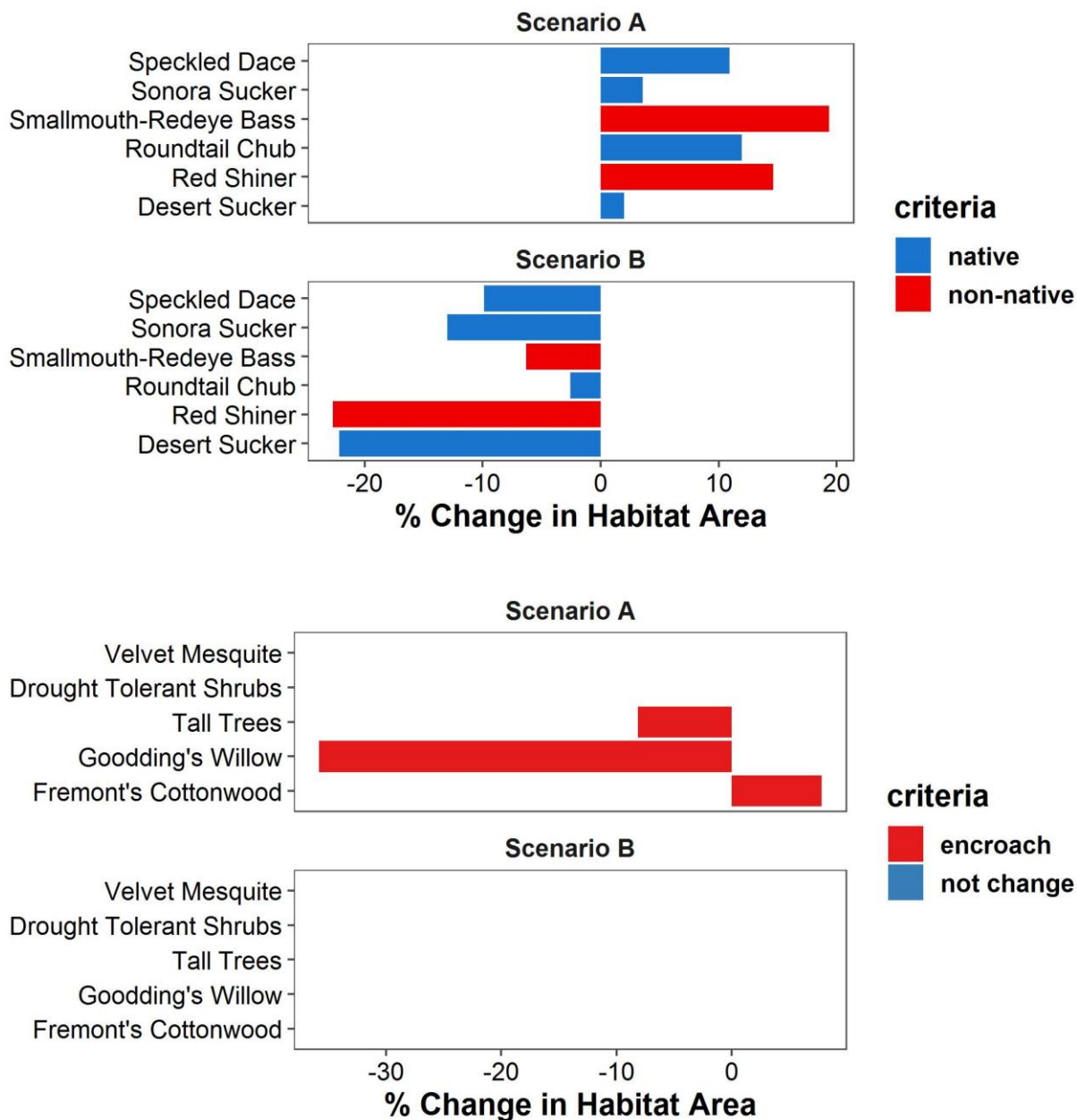


Figure 8. Overall percent change in habitat area for fish (top) and vegetation (bottom) with the category of change. Plant groups with missing bars had no change in habitat area.

Both scenarios I evaluated with the REFDSSV2 had their own benefits and tradeoffs. Gains in native fish habitat in Scenario A were largely overshadowed by gains in non-native fish

habitat and vegetation loss and encroachment. In Scenario B vegetation was unaffected and non-native fish habitat declined, but native fish habitat was greatly reduced during the summer.

Scenario A evaluated the impacts of reducing high flows. The impacts of reducing peak flows in Scenario A are consistent with the literature on fish and floodplain vegetation. Reducing peak flows improved habitat for fish species that do not tolerate high flows. Non-native species in particular have been shown to increase when high flows are reduced because these high flows are frequently limiting factors (Marchetti and Moyle, 2001; Lytle and Poff, 2004; Kiernan et al., 2012). Floodplain vegetation are particularly sensitive to reduced high flows because their traits are adapted to those flows (Lytle and Poff, 2004; Tonkin et al., 2018). Vegetation have been observed moving closer to the channel when high flows decrease because of lack of fluvial disturbance and access to groundwater (Miller et al., 2013).

Scenario B evaluated the impacts of reducing flows during periods of baseflow. Floodplain vegetation were unaffected by this scenario because high flows remained intact. Tall trees and drought tolerant shrubs favored areas with low *EP* values. These *EP* values did not change in this scenario. Vegetation in the study area are able to handle periods of drought and declines in baseflow as long as roots can reach the water table (Paretti et al., 2018). Fish habitat declined with baseflow reductions. Habitat availability is already limited during times of low flow (Paretti et al., 2018), so reducing this habitat further lead to a large proportion of habitat lost.

The REFDSSV2 does not try to identify the best alternative flow scenario; the REFDSSV2 supplies outputs that support decision making. Examining monthly fish habitat change gives information about how habitat change or maintenance is balanced throughout the year in a given scenario. An increase in an individual's habitat for a single season could be undermined by a drastic reduction in habitat in subsequent seasons. A challenge of comparing fish and vegetation response is that they typically operate on different life cycles (Rivaes et al., 2017). Overall habitat change as shown in Figure 8 provides a means for comparing annual average change in fish habitat and long term changes in vegetation habitat. Vegetation

movement is an important factor to consider given vegetation's influence on channel width, habitat availability, and geomorphic processes (Javernick and Bertoldi, 2019).

The relative importance of the outputs should be determined by the stakeholders. Factors to consider include the value of native versus non-native species, the balance of vegetation maintenance and movement, the presence of special status or invasive species, and the timing of habitat changes. Information that is not captured by the REFDSSV2 should also be considered. For example species interactions like competition and predation could affect habitat availability.

#### 5.1 Limitations of REFDSSV2

The REFDSSV2 is inherently limited by the quality of the 2D hydrodynamic modeling results. These results depend on data availability, model resolution, and calibration standards. Sparse topological data can limit the resolution of habitat modeling. Incomplete stage-discharge information can limit model calibration. Without well-calibrated high resolution 2D hydrodynamic models, measures of habitat availability are limited.

HSC was limited to hydraulic conditions and substrate type. This binary habitat preference information does not necessarily reflect habitat quality (Railsback et al., 2003). For instance, an abundance of marginal habitat could obscure the impacts of a flow scenario on a species. Conversely, suitable habitat could exist outside the preferred range of conditions (Railsback et al., 2003). Similarly, the logistic regression equations for vegetation return probabilities of occurrence, so vegetation could occur outside areas included in habitat availability quantities. A low to moderate probability of occurrence does not equate to the absence of vegetation nor does a high probability of occurrence guarantee the vegetation will grow there.

The REFDSSV2 is only able to quantify habitat availability; it cannot describe actual habitat utilization. Habitat utilization depends on complex ecological processes that are not captured by the tool. Additionally, the lack of population component limits its ability to describe



carrying capacity, recruitment, and demographics. Although it was not demonstrated in the Verde River case study, the REFDSSV2 is capable of modeling multiple life stages for fish. Currently, relationships between these life stages that determine habitat accessibility cannot be specified, but this feature can be added in the future.

## 5.2 Habitat Restoration and Protection

A secondary application of the REFDSSV2 is identifying areas that should be targeted for restoration or protection efforts. Effectively allocating the limited resources for these projects requires understanding current and potential habitat suitability. Habitat can be evaluated for historic or proposed conditions at multiple reaches using the REFDSSV2.

In order to protect habitat for special status or high value species, managers need to identify where their habitat is located. Probability of occurrence and fish habitat maps can be used to identify areas in a reach that produce suitable conditions. For example, Wyman et al. (2018) combined fine-scale habitat modeling with fish telemetry data to identify specific locations of spawning habitat for Green Sturgeon in reaches they were known to use. When alternative flow scenarios are being considered, the REFDSSV2 can be used to differentiate between vegetation habitat that will persist and habitat that will be vulnerable through a proposed change. Habitat area-discharge curves and min10 time series provide information about changes in fish habitat availability between reaches. Understanding habitat availability helps managers prioritize areas that could benefit the most from protection.

Successfully restoring a species' habitat or population depends on choosing the right location and conditions. For example, the fine-scale information provided by the REFDSSV2 can be used to identify where conditions are most suitable for native vegetation to be replanted or invasive species to be removed. Similar information has been used to decide which reaches were worth restoring (Julian et al., 2016). Length-normalized habitat availability for fish can be compared reach-to-reach through area-discharge curves and min10 time series. This information may be used to decide where fish should be stocked. Restoration may also involve

altering flows to support a population. Metrics of change from the *scenario comparison* module can be used to determine which flow scenarios maintain the most habitat in a specific reach or which reaches have the most habitat under a new flow scenario. Choosing locations and flow alterations based on habitat availability supports the long-term success of those restoration efforts.

## **6 Conclusions**

Environmental flow planning involves evaluating tradeoffs and making compromises. Existing tools have tackled tradeoff evaluation in different ways. Holistic tools incorporate factors beyond instream and floodplain habitat at the expense of high-resolution spatial analysis. Conversely, tools with a strong spatial basis separate instream and floodplain modules or do not include vegetation at all.

The REFDDSSV2 was designed to evaluate tradeoffs of both instream and floodplain habitats in a variety of alternative flow scenarios and environments using simple, open-ended metrics and high-resolution habitat analysis. This tool aims to address the short comings of existing tools including the REFDDSSV1. Mapping habitat using 2D hydrodynamic models allows impacts on those habitats to be described quantitatively. This high-resolution habitat information also creates the opportunity to use the REFDDSSV2 for restoration and protection efforts. Changes in habitat availability were described through percent change in habitat area, balance of native and nonnative fish species, and potential vegetation movement. Users can adapt this tool to their study reaches through their choice of inputs. Analysis can be conducted for multiple species, life stages, and vegetation guilds. The long-term impacts of hydrologic alteration including withdrawals, reservoir operations, climate change, and other changes to the flow regime can be evaluated.

A limitation of the REFDDSSV2 is its inability to describe habitat utilization. Currently, there is no demographic component, so processes including recruitment, competition, and

carrying capacity cannot be explicitly represented. Habitat availability does not mean that it will be or can be utilized by a species or group.

Development of the REFDSSV2 will continue in the future. The intention is to create a web-based platform available for the US Forest Service and stakeholders to use. It will feature a robust graphical user interface and expanded capabilities. The accessibility and adaptability of this tool will be maintained by keeping the open-source structure. Proposed additions include calculations to account for competition, recruitment, and life cycles.

## References

- Ahmadi-Nedushan, B., St-Hilaire, A., Bérubé, M., Robichaud, É., Thiémonge, N., Bobée, B., 2006. A review of statistical methods for the evaluation of aquatic habitat suitability for instream flow assessment. *River Res. Appl.* 22, 503–523. <https://doi.org/10.1002/rra.918>
- Alafifi, A.H., Rosenberg, D.E., 2020. Systems modeling to improve river, riparian, and wetland habitat quality and area. *Environmental Modelling & Software* 126, 104643. <https://doi.org/10.1016/j.envsoft.2020.104643>
- Arthington, A.H., Baran, E., Brown, C.A., Dugan, P., Halls, A.S., King, J.M., Minte-Vera, C.V., Tharme, R.E., Welcome, R.L., 2007. Water requirements of floodplain rivers and fisheries: existing decision support tools and pathways for development.
- Baird, K.J., Stromberg, J.C., Maddock, T., 3rd, 2005. Linking riparian dynamics and groundwater: an ecohydrologic approach to modeling groundwater and riparian vegetation. *Environ. Manage.* 36, 551–564. <https://doi.org/10.1007/s00267-004-0181-z>
- Barrett, P.J., Maughan, O.E., 1995. Spatial Habitat Selection of Roundtail Chub (*Gila robusta*) in Two Central Arizona Streams. *Southwest. Nat.* 40, 301–307.
- Bestgen, K.R., Propst, D.L., 1989. Distribution, Status, and Notes on the Ecology of *Gila robusta* (Cyprinidae) in the Gila River Drainage, New Mexico. *Southwest. Nat.* 34, 402–412. <https://doi.org/10.2307/3672169>
- Bovee, K.D., 1986. Development and evaluation of habitat suitability criteria for use in the instream flow incremental methodology (No. FWS/OBS-86/7). USDI Fish and Wildlife Service.
- Brouder, M.J., 2000. Life History and Ecology of the Roundtail Chub, *Gila Robusta*, from Two Streams in the Verde River Basin. Arizona Game and Fish Department, Research Branch.
- Cunnane, C., 1978. Unbiased plotting positions — A review. *J. Hydrol.* 37, 205–222. [https://doi.org/10.1016/0022-1694\(78\)90017-3](https://doi.org/10.1016/0022-1694(78)90017-3)
- Diehl, R.M., Merritt, D.M., Wilcox, A.C., Scott, M.L., 2017. Applying Functional Traits to Ecogeomorphic Processes in Riparian Ecosystems. *Bioscience* 67, 729–743. <https://doi.org/10.1093/biosci/bix080>
- Dunbar, M.J., Alfredsen, K., Harby, A., 2012. Hydraulic-habitat modelling for setting environmental river flow needs for salmonids. *Fish. Manag. Ecol.* 19, 500–517. <https://doi.org/10.1111/j.1365-2400.2011.00825.x>
- González, E., González-Sanchis, M., Comín, F.A., Muller, E., 2012. Hydrologic thresholds for riparian forest conservation in a regulated large Mediterranean river. *River Res. Appl.* 28, 71–80. <https://doi.org/10.1002/rra.1436>
- Hickey, J.T., 2017. HEC-EFM Ecosystem Functions Model. Quick Start Guide. U.S. Army Corp of Engineers.

- Hickey, J.T., Huff, R., Dunn, C.N., 2015. Using habitat to quantify ecological effects of restoration and water management alternatives. *Environmental Modelling & Software* 70, 16–31. <https://doi.org/10.1016/j.envsoft.2015.03.012>
- Hijmans, R.J., Van Etten, J., Cheng, J., Mattiuzzi, M., Sumner, M., Greenberg, J.A., Lamigueiro, O.P., Bevan, A., Racine, E.B., Shortridge, A., Others, 2019. raster: Geographic Data Analysis and Modeling.
- Javernick, L., Bertoldi, W., 2019. Management of vegetation encroachment by natural and induced channel avulsions: A physical model. *River Res. Appl.* 35, 1257–1268. <https://doi.org/10.1002/rra.3494>
- Julian, D.W., Hickey, J.T., Fields, W.L., Ostadrahimi, L., Maher, K.M., Barker, T.G., Hatfield, C.L., Lutz, K., Marks, C.O., Sandoval-Solis, S., Lund, J.R., 2016. Decision Support System for Water and Environmental Resources in the Connecticut River Basin. *Journal of Water Resources Planning and Management* 142. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000538](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000538)
- Kuhn, M., 2012. The caret package. R Foundation for Statistical Computing, Vienna, Austria. URL <https://cran.r-project.org/package=caret>.
- Lamoureux, N., Hauer, C., Stewardson, M.J., LeRoy Poff, N., 2017. Physical Habitat Modeling and Ecohydrological Tools, in: Horne, A.C., Webb, J.A., Stewardson, M.J., Richter, B., Acreman, M. (Eds.), *Water for the Environment*. Academic Press, pp. 265–285. <https://doi.org/10.1016/B978-0-12-803907-6.00013-9>
- Lytle, D.A., Merritt, D.M., 2004. HYDROLOGIC REGIMES AND RIPARIAN FORESTS: A STRUCTURED POPULATION MODEL FOR COTTONWOOD. *Ecology* 85, 2493–2503. <https://doi.org/10.1890/04-0282>
- Lytle, D.A., Merritt, D.M., Tonkin, J.D., Olden, J.D., Reynolds, L.V., 2017. Linking river flow regimes to riparian plant guilds: a community-wide modeling approach. *Ecol. Appl.* 27, 1338–1350. <https://doi.org/10.1002/eap.1528>
- Mahoney, J.M., Rood, S.B., 1998. Streamflow requirements for cottonwood seedling recruitment—An integrative model. *Wetlands* 18, 634–645. <https://doi.org/10.1007/BF03161678>
- Marsili-Libelli, S., Giusti, E., Nocita, A., 2013. A new instream flow assessment method based on fuzzy habitat suitability and large scale river modelling. *Environmental Modelling & Software* 41, 27–38. <https://doi.org/10.1016/j.envsoft.2012.10.005>
- Merritt, D.M., Scott, M.L., LeROY POFF, N., Auble, G.T., Lytle, D.A., 2010. Theory, methods and tools for determining environmental flows for riparian vegetation: riparian vegetation-flow response guilds: Riparian vegetation-hydrologic models. *Freshw. Biol.* 55, 206–225. <https://doi.org/10.1111/j.1365-2427.2009.02206.x>
- Milhous, R.T., Wegner, D.L., Waddle, T., 1984. User's Guide to the Physical Habitat Simulation System (PHABSIM). Department of the Interior, U.S. Fish and Wildlife Service.

- Morrison, R.R., Stone, M.C., 2014. Spatially implemented Bayesian network model to assess environmental impacts of water management. *Water Resour. Res.* 50, 8107–8124. <https://doi.org/10.1002/2014WR015600>
- Nelson, J.M., Shimizu, Y., Abe, T., Asahi, K., Gamou, M., Inoue, T., Iwasaki, T., Kakinuma, T., Kawamura, S., Kimura, I., Kyuka, T., McDonald, R.R., Nabi, M., Nakatsugawa, M., Simões, F.R., Takebayashi, H., Watanabe, Y., 2016. The international river interface cooperative: Public domain flow and morphodynamics software for education and applications. *Adv. Water Resour.* 93, 62–74. <https://doi.org/10.1016/j.advwatres.2015.09.017>
- Nemec, Z., 2018. Development and evaluation of habitat suitability criteria for native fishes and assessment of the relationship among riparian areas and stream macrohabitats type and fish presence in four central Arizona streams (Masters). The University of Arizona.
- Paretti, N., Brasher, A.M.D., Pearlstein, S.L., Skow, D.M., Gungle, B.W., Garner, B.D., 2018. Preliminary synthesis and assessment of environmental flows in the Middle Verde River Watershed, Arizona (No. 2017-5100), Scientific Investigations Report. U.S. Geological Survey. <https://doi.org/10.3133/sir20175100>
- Paroz, Y.M., Propst, D.L., Stefferud, J.A., 2006. Long-term monitoring of fish assemblages in the Gila River drainage, New Mexico. New Mexico Department of Game and Fish, Santa Fe, NM.
- Poff, N.L., Matthews, J.H., 2013. Environmental flows in the Anthropocene: past progress and future prospects. *Current Opinion in Environmental Sustainability* 5, 667–675. <https://doi.org/10.1016/j.cosust.2013.11.006>
- Poff, N.L., Richter, B.D., Arthington, A.H., Bunn, S.E., Naiman, R.J., Kendy, E., Acreman, M., Apse, C., Bledsoe, B.P., Freeman, M.C., Henriksen, J., Jacobson, R.B., Kennen, J.G., Merritt, D.M., O'Keefe, J.H., Olden, J.D., Rogers, K., Tharme, R.E., Warner, A., 2010. The ecological limits of hydrologic alteration (ELOHA): a new framework for developing regional environmental flow standards. *Freshw. Biol.* 55, 147–170. <https://doi.org/10.1111/j.1365-2427.2009.02204.x>
- Railsback, S.F., Stauffer, H.B., Harvey, B.C., 2003. WHAT CAN HABITAT PREFERENCE MODELS TELL US? TESTS USING A VIRTUAL TROUT POPULATION. *Ecol. Appl.* 13, 1580–1594. <https://doi.org/10.1890/02-5051>
- Richter, B., Orr, S., 2017. Chapter 16 - Water Budgets to Inform Sustainable Water Management, in: Horne, A.C., Webb, J.A., Stewardson, M.J., Richter, B., Acreman, M. (Eds.), *Water for the Environment*. Academic Press, pp. 347–360. <https://doi.org/10.1016/B978-0-12-803907-6.00016-4>
- Rinne, J.N., 1992. Physical habitat utilization of fish in a Sonoran Desert stream, Arizona, southwestern United States. *Ecol. Freshw. Fish* 1, 35–41. <https://doi.org/10.1111/j.1600-0633.1992.tb00005.x>
- Rivaes, R., Boavida, I., Santos, J.M., Pinheiro, A.N., Ferreira, T., 2017. Importance of considering riparian vegetation requirements for the long-term efficiency of environmental flows. <https://doi.org/10.5194/hess-2017-57>

- Steinschneider, S., Bernstein, A., Palmer, R., Polebitski, A., 2014. Reservoir Management Optimization for Basin-Wide Ecological Restoration in the Connecticut River. *Journal of Water Resources Planning and Management* 140, 04014023.  
[https://doi.org/10.1061/\(asce\)wr.1943-5452.0000399](https://doi.org/10.1061/(asce)wr.1943-5452.0000399)
- Tarlock, A.D., Tippy, R., 1969. Wild and Scenic Rivers Act of 1968. *Cornell Law Rev.* 55, 707.
- Tonkin, J.D., Olden, J.D., Merritt, D.M., Reynolds, L.V., Rogosch, J.S., Lytle, D.A., 2020. Designing flow regimes to support entire river ecosystems. *bioRxiv*.  
<https://doi.org/10.1101/2020.01.09.901009>
- U.S. Fish and Wildlife Service, 2009. Sonora Sucker (*Catostomus insignis*).
- U.S. Forest Service, 2004. Verde Wild and Scenic River comprehensive river management plan final environmental assessment. USDA.
- Waddle, T.J., Bovee, K.D., 2010. USGS Open-File Report 2009-1272: Environmental Flow Studies of the Fort Collins Science Center—Cherry Creek, Arizona [WWW Document]. U.S. Geological Survey (USGS). URL <https://pubs.usgs.gov/of/2009/1272/> (accessed 5.17.20).
- Waters, S., 2017. User manual for the DRIFT DSS software.
- Wyman, M.T., Thomas, M.J., McDonald, R.R., Hearn, A.R., Battleson, R.D., Chapman, E.D., Kinzel, P., Minear, J.T., Mora, E.A., Nelson, J.M., Pagel, Klimley, A.P., 2018. Fine-scale habitat selection of green sturgeon (*Acipenser medirostris*) within three spawning locations in the Sacramento River, California. *Can. J. Fish. Aquat. Sci.* 75, 779–791.  
<https://doi.org/10.1139/cjfas-2017-0072>
- Yao, W., Bui, M.D., Rutschmann, P., 2018. Development of eco-hydraulic model for assessing fish habitat and population status in freshwater ecosystems: eco-hydraulic model in freshwater ecosystems. *Ecohydrol.* 11, e1961. <https://doi.org/10.1002/eco.1961>

## Appendix A: REFDSSV2 Basecode

### pre.processing.R (Module 1)

```
# This script will house the pre-processing options and functionality
# This script will process 2D hydraulic modeling results
# This script will rasterize substrate maps if applicable
# Last edited by Elaina Passero on 04/20/20

# Load required packages
packages <-
c("SDMTools", "sp", "raster", "rgeos", "rgdal", "sf", "spatstat", "spdep", "tidyverse", "rasterVis",

"ggplot2", "data.table", "dplyr", "plotly", "spex", "stars", "igraph", "deldir", "hydroTSM",
"rlist")

# Check to see if each is installed, and install if not.
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}

# load the installed libraries in the packages list
lapply(packages, library, character.only=TRUE)

## Primary Inputs
wd <- "C:/Users/epassero/Desktop/VRDSS/verde-refdss/" # Project working directory
#wd <- "/Users/Morrison/Documents/Active Research Projects/Verde REFDSS/verde-
refdss/" # Set path to local repository
setwd(wd)
hab_mets <- list("Depth", "Velocity") #, "Velocity", "WaterSurfaceElevation")
#Variables from iRIC calculation result used for habitat analysis - case sensitive!
reach_name <- "USBeasley1" # Reach of interest
model_run <- "2.12.20" # Should match end of name of folder with results
dis_unit <- "cms" #units of dischargee

# Yes- external rasters or No- rasterize iRIC results. Inputs required if No.
LoadExternal <- "No"; if(LoadExternal=="No"){
  skip_num <- 0 # number of rows to skip when reading in raw outputs
  x_loc <- "X" # field name of X coordinate in model outputs - case sensitive
  y_loc <- "Y" # field name of y coordinate in model outputs - case sensitive
  dem <- "BeasleyUS_0.25m_20191108.tif" # Name of dem: VerdeBeasley1Elev.tif,
smrf_dem_v241.tif, braidallpts_dem.tif, GilaMGnd.tif
  # Does the resolution of the rasters need to be manually set? If No, dem
  resolution will be used.
```



```

setRes <- "Yes"; if(setRes=="Yes"){
  res <- c(0.5,0.5)} # resolution of rasters if they need to be manually set
}

## Options - If set to No, inputs are not required for option
# Yes or No. Choose whether or not to check substrate conditions as part of suitable
habitat
CheckSub <- "Yes"; if(CheckSub=="Yes"){
  sub_name <- "BeasleyUS_SedThiessenPoly1Dissolved" # shapefile name with no
extension BeasleyUS_SedThiessenPoly1Dissolved
  sub_field <- "ParticalSi"} # name of field in substrate map containing substrate
type info; ParticalSi or substrate

### End of User Inputs ###

### Begin Processing ###
reach_run <- paste(reach_name,"_",model_run,sep="")
reach_wd <- paste(wd,"reaches","/",reach_run,"/",sep = "")
rasterOptions(overwrite = TRUE,tmpdir = paste(reach_wd,"temp/",sep=""))
pre_outputs <- list()

if(LoadExternal == "No"){
  ## Format result CSVs and get list of discharges
  source("get.results.R")
  temp_list <- get.results(reach_wd,skip_num,dis_unit)
  delim_list <- temp_list$delim_list
  modeled_q <- temp_list$modeled_q
  rm(temp_list)

  ## Convert iRIC outputs to rasterBricks by variable
  source("iric.process.smr.R")
  out_val_rast <- list()
  out_val_rast <- lapply(hab_mets, function(m)
iric.process.smr(m,delim_list,reach_wd,dem,x_loc,y_loc,setRes,res))
  names(out_val_rast) <-hab_mets
  rm(delim_list)
} else{

  ## Load in external rasterBricks and discharges
  source("load.cherry.R")
  out_val_rast <- load.cherry(reach_wd,Check0Flow)
  names(out_val_rast)<-c(hab_mets,"modeled_q")
  modeled_q <- out_val_rast$modeled_q

```

```

    out_val_rast[length(out_val_rast)]<-NULL
  }

pre_outputs$out_val_rast <- out_val_rast
pre_outputs$modeled_q <- modeled_q

## Load substrate
if(CheckSub == "Yes"){
  base_rast <- out_val_rast[[1]][[1]] # will be overwritten during rasterization -
  provides setup
  sub_map <- readOGR(dsn=paste(wd,"reaches", "/", reach_run, sep=""),layer=sub_name)
  # read in substrate shapefile
  rast_sub_map <-
  rasterize(sub_map,base_rast,field=sub_map@data[[sub_field]],update=TRUE)

  pre_outputs$rast_sub_map <- rast_sub_map
}

# save outputs for internal use
list.save(pre_outputs,file=paste(reach_wd,"dss_outputs/","internal/",reach_run,"_pre
_outputs.rdata",sep=""))

```

## get.results.R

```

# This function will pull in iRIC output results and format them to be rasterized
# Last updated by Elaina Passero 1/13/20

# Requirements for running this code:
# Delimited files must be named with their discharge in the name. Discharge values
# should be numeric only.

get.results <- function(reach_wd,skip_num,dis_unit){

  # DEM and iRIC calculation results must be in their own folder
  tempwd <- paste(reach_wd,"model_outputs", "/", sep = "")
  # get names of iric output csv files in the working directory
  iric_results <- list.files(path=tempwd,pattern = ".csv") # model_outputs should not
  contain any other CSVs

  # Create empty list to contain rasters
  hold_list <- list()
  delim_list <- list()

```

```

# Create list of modeled Qs
modeled_q <- c()
modeled_q <- parse_number(iric_results)

delim_list <- lapply(iric_results, function(i){
  df <- fread(file=paste(tempwd,i,sep = ""),
              skip = skip_num,header = TRUE, sep=",",check.names=TRUE,data.table = FALSE)
  cn <- colnames(df)
  colnames(df) <- word(gsub("[.]", " ",cn)) # removes periods from DF column names
  and leaves only first word
  return(df)
})

names(delim_list) <- paste(dis_unit,"_",modeled_q,sep="")

hold_list$delim_list <- delim_list
hold_list$modeled_q <- modeled_q
return(hold_list)
} # end of function

```

## iric.process.smr.R

```

# Adapted from IRIC_processing_v1 by Julian Scott 01/30/19
# Function: Processes iRIC output by discharge and variable. Returns list of
rasters.
# Last edited by Elaina Passero on 02/11/20

iric.process.smr <- function(m,delim_list,reach_wd,dem,x_loc,y_loc,setRes,res) {
  # read in elevation surface from the working directory
  smr_elev <- raster(paste(reach_wd,"model_outputs","/",dem,sep = ""))
  # create an empty raster with the extent, resolution, and projection of the DEM.
  e <- extent(smr_elev)
  if(setRes == "No"){
    res <- res(smr_elev)
  }
  set_crs <- crs(smr_elev)
  r <- raster(x = e,resolution = res,crs = set_crs)

  # Transfer values from iric output i to the cells of raster r, but only when the
  cell is inundated (!= 0)

```

```

# If multiple points from the iric output are within a cell of r, the mean points
is used for the cell value.
depth <- grep("depth",names(delim_list[[1]]),ignore.case = TRUE,value = TRUE)
smr_q_s <- stack(lapply(delim_list,function(i) rasterize(x = i[,c(x_loc,y_loc)],y
= r,
field = ifelse(i[,depth]
== 0, NA, i[,m]),fun = mean)))
proj4string(smr_q_s) <- set_crs
# Resample raster i using bilinear interpolation to fill in those cells in r_i
that did not have a cell value due to no point overlap
smr_q_s <- raster::projectRaster(from=smr_q_s,to=smr_elev,method = 'bilinear')
# Returns Brick of rasters
return(smr_q_s)
} # end function

```

## fish.processing.R (Module 2)

```

# This script will house the fish processing options and functionality
# This script will produce area-lookup tables by species for fish
# Last edited by Elaina Passero on 6/3/20

# Load required packages
packages <-
c("SDMTools","sp","raster","rgeos","rgdal","sf","spatstat","spdep","tidyverse","raste
rVis",

"ggplot2","ggthemes","data.table","dplyr","plotly","spex","stars","igraph","deldir","
hydroTSM","rlist")
# Check to see if each is installed, and install if not.
if (length(setdiff(packages, rownames(installed.packages())) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}
# load the installed libraries in the packages list
lapply(packages,library,character.only=TRUE)

### Begin Inputs ###
fish_tag <- "d2" # distinguish file name of area discharge curves

## Primary Inputs
wd <- "C:/Users/epassero/Desktop/VRDSS/verde-refdss/" # Project working directory
setwd(wd)

```

```

hab_mets <- list("Depth","Velocity") #Variables from iRIC calculation result used for
habitat analysis - case sensitive
species_list <- c("Desert_Sucker","Roundtail_Chub","Sonora_Sucker","Speckled_Dace",
                  "Red_Shiner","Smallmouth-Redeye_Bass") # it's recommended that
species be listed natives first in alphabetical order
native_list <- c("Desert_Sucker","Roundtail_Chub","Speckled_Dace","Sonora_Sucker") #
list of native fish species in species_list.
lifestages <- list("adult") #lifestages from oldest to youngest; must match order in
HSC table
reach_name <- "USBeasley1" # Reach of interest
model_run <- "2.12.20" # Should match end of name of folder with results
dis_unit <- "cms" #units of discharge

## Secondary Inputs - Use only if switching between projects
Check0Flow <- "No" # Yes- Calculate max area for 0-flow scenario and interpolate
below min modeled Q. No - 0 flow = 0 area.
if(Check0Flow=="Yes"){
  depth0Flow <- "d_cfs_0m" # raster name for 0 flow depth
}# end of internal rasterization inputs;

## Options - If set to No, inputs are not required for option
# Yes or No. Choose whether or not to check substrate conditions as part of suitable
habitat
CheckSub <- "Yes"

# Yes or No. Choose whether or not to remove isolated (single cell) habitat patches
RemoveIslands <- "Yes"; if(RemoveIslands=="Yes"){
  island_size <- 2} # number of raster cells that is considered too small of a
habitat patch

# Yes or No. Choose whether or not to normalize habitat area by reach length
NormalizeByL <- "Yes"; if(NormalizeByL=="Yes"){
  reach_length <- 0.5
  length_unit <- "km"}

# Yes or No. Export Rasters of Habitat by Discharge per lifestage and species
ExportRast <- "Yes"

# Yes or No. Export habitat area-lookup tables
ExportTable <- "Yes"

### End User Inputs ###

```

```

### Begin Processing ###
reach_run <- paste(reach_name, "_", model_run, sep="")
reach_wd <- paste(wd, "reaches", "/", reach_run, "/", sep = "")

output_name <-
load(file=paste(reach_wd, "dss_outputs/", "internal/", reach_run, "_pre_outputs.rdata", sep="")) # load fish outputs
eval(parse(text=paste("pre_outputs=", output_name)))

out_val_rast <- pre_outputs$out_val_rast
modeled_q <- pre_outputs$modeled_q

if(CheckSub == "Yes"){
  rast_sub_map <- pre_outputs$rast_sub_map
}

##### Run for all species #####
fish_outputs <- list()
fish_outputs <- lapply(species_list, function(species){ # builds tables and maps for
all species in list

  fish <- species

  ## Reclassify Bricks with hydraulic and substrate HSC by lifestage
  source("find.hsc.R"); source("bricks.rc.R"); source("by.substrate.R");
source("find.sub.R"); source("remove.islands.R")

  hsc_all_spec <-
fread(paste(reach_wd, "habitat_info", "/", reach_name, "_hsc", ".csv", sep = ""),
      header=TRUE, sep=",", data.table = FALSE)

  hsc_all_ages <- find.hsc(hsc_all_spec, species) # extract HSC for single species
  good_hab_list <- lapply(lifestages, function(a)
bricks.rc(a, out_val_rast, hsc_all_ages, hab_mets))
  names(good_hab_list) <- lifestages # list of Bricks by lifestage

  if(CheckSub == "Yes"){
    sub_all_spec <-
fread(paste(reach_wd, "habitat_info", "/", reach_name, "_substrate", ".csv", sep=""),
      header=TRUE, sep = ",", data.table = FALSE) # load substrate
requirements
    sub_all_ages <- find.sub(sub_all_spec, species) # extract substrate requirements
for single species

```

```

    good_hab_list <- lapply(lifestages, function(a) by.substrate(a, good_hab_list,
sub_all_ages,rast_sub_map,fish))
    names(good_hab_list) <- lifestages
  } # end of if statement

  if(RemoveIslands == "Yes"){
    good_hab_list <- lapply(lifestages, function(a)
remove.islands(a,good_hab_list,RemoveIslands,island_size))
    names(good_hab_list) <- lifestages
  }

  ## Total available habitat area by lifestage
  source("total.area.R")
  area_look_tab <- lapply(lifestages, function(a)
total.area(a,good_hab_list,modeled_q,NormalizeByL,

reach_length,hab_mets,ExportTable,fish,reach_run))
  names(area_look_tab) <- lifestages

  ## Order rasters of total available habitat by modeled discharge
  source("order.by.q.R")
  out_by_q <- lapply(lifestages, function(a)
order.by.q(a,good_hab_list,modeled_q,ExportRast,reach_wd,fish,reach_run))
  names(out_by_q) <- lifestages
  rast_by_q <- unlist(lapply(lifestages, function(a){out_by_q[[a]][[1]]}))
  sort_q <- unlist(lapply(lifestages, function(a){out_by_q[[a]][[2]]}))

  # condense outputs into a single list
  fish_outputs$area_look_tab <- area_look_tab
  fish_outputs$rast_by_q <- rast_by_q

  # save outputs for internal use

list.save(fish_outputs,file=paste(reach_wd,"dss_outputs/","internal/",reach_run,"_",fish,
"_fish_outputs.rdata",sep=""))

  return(fish_outputs)
}) # end of species list function

names(fish_outputs) <- species_list

# create plots of the habitat area-discharge curves

```

```

source("plot.a.q.curve.R"); source("plot.a.q.curve.in.chan.R")
area_look_plots <- lapply(lifestages, function(a)
plot.a.q.curve(fish_outputs,species_list,a,NormalizeByL,fish_tag,reach_run,native_list))
in_chan_area_look_plots <- lapply(lifestages, function(a)
plot.a.q.curve.in.chan(fish_outputs,species_list,a,NormalizeByL,fish_tag,reach_run,native_list))

```

## find.hsc.R

```

# This function creates an HSC table for the species of interest
# Last edited by Elaina Passero on 4/26/19

find.hsc <- function(hsc_all_spec,species){
  life_title <- unlist(hsc_all_spec[,1])
  pos <- grep(species,life_title,ignore.case = TRUE)
  hsc_all_ages <- hsc_all_spec[pos,] # HSC for single species
  return(hsc_all_ages)
}

```

## find.sub.R

```

# This function pulls substrate requirements for a single species from the
substrate table
# Last edited by Elaina Passero on 4/20/20

find.sub <- function(sub_all_spec,species){
  life_title <- unlist(sub_all_spec[,1])
  pos <- grep(species,life_title,ignore.case = TRUE)
  sub_all_ages <- sub_all_spec[pos,] # HSC for single species
  return(sub_all_ages)
}

```

## bricks.rc.R

```

# Function: Reclassifies rasterStacks with habitat suitability criteria.
# Outputs RasterBrick with layers of suitable cells by discharge
# Last edited by Elaina Passero on 6/6/20

### Begin function ###
bricks.rc <- function(a,out_val_rast,hsc_all_ages,hab_mets){

```



```

pos <- grep(a,hsc_all_ages,ignore.case = TRUE)
hsc <- hsc_all_ages[pos,] # HSC for current lifestage

accept_rast <- setValues(out_val_rast[[1]],0) # creates rasterBrick of all 0's
where acceptable cells will be added to by discharge

for(m in 1:length(hab_mets)){
  hm <- grep(hab_mets[m],names(hsc),ignore.case = TRUE) # position of habitat
metric in HSC table
  p_min <- min(hm)
  p_max <- max(hm) # set positions of min and max values for habitat metric
  s_depth <- c(-Inf,hsc[,p_min],NA, hsc[,p_min],hsc[,p_max],1,
hsc[,p_max],Inf,NA)
  rc_dep_mat <- matrix(s_depth, ncol=3, byrow=TRUE)
  rc_dep_rast <- reclassify(out_val_rast[[m]], rc_dep_mat,right=FALSE)
  accept_rast <- accept_rast + rc_dep_rast # any value + NA yields NA; cells
either number of habitat metrics or NA
} # end of for loop
return(accept_rast) # Returns Brick of suitable cells by discharge
} # end of function

```

## by.substrate.R

```

# This script is used to get suitable habitat that meets substrate type
requirements.
# Last edited by Elaina Passero on 4/20/20

### Begin Function ###
by.substrate <- function(a, good_hab_list, sub_all_ages,rast_sub_map,fish){
  good_rast <- good_hab_list[[a]] # raster of suitable habitat for current lifestage

  pos <- grep(a,sub_all_ages,ignore.case = TRUE)
  sub_req <- sub_all_ages[pos,] # Substrate requirement for current lifestage

  s_sub <- c(-Inf,sub_req[,2],NA, sub_req[,2],sub_req[,3],1, sub_req[,3],Inf,NA)
  rc_sub_mat <- matrix(s_sub, ncol=3, byrow=TRUE)
  rc_sub_map <- reclassify(rast_sub_map, rc_sub_mat,right=FALSE) # reclassify
substrate map based on substrate criteria

  writeRaster(rc_sub_map,filename =
paste(reach_wd,"dss_outputs/",fish,"_",a,"_sub.tif",sep=""),format="GTiff",overwrite
=TRUE)

```

```

# Mask
by_sub_brick <- mask(good_rast,rc_sub_map,inverse=TRUE,maskvalue=1,updatevalue=NA)
# if cells not covered by acceptable substrate or are NA, they are set to NA
return(by_sub_brick)
}

```

## remove.islands.R

```

# This function removes islands (isolated, small habitat areas) from habitat area
calculation and maps
# Last edited by Elaina Passero on 10/18/19

remove.islands <- function(a,good_hab_list,RemoveIslands,island_size){
  if(nlayers(good_hab_list[[a]])==0){
    b <- brick(good_hab_list[[a]]) # ensures a single raster layer will "unstack"
  } else{
    b <- good_hab_list[[a]]
  }
  ## remove islands from rasters using clumping
  if(RemoveIslands == "Yes"){
    u <- unstack(b)
    f <- lapply(u, function(e){
      hab_clump <- clump(e,directions=4) # group raster's into clumps using rook's
rules
      clump_freq <- as.data.frame(freq(hab_clump)) # create df of frequencies
      exclude_id <- clump_freq$value[which(clump_freq$count<=island_size)] # find
IDs of isolated cells
      e[hab_clump %in% exclude_id] <- NA # Assign NA to all isolated cells
      return(e)
    })
    b <- brick(f)
  } # end of if statement
  return(b)
}

```

## total.area.R

```

# Calculates total available habitat area by lifestage
# Last edited by Elaina Passero on 2/19/20

```

```

total.area <-
function(a,good_hab_list,modeled_q,NormalizeByL,reach_length,hab_mets,ExportTable,fi
sh,reach_run){
  b <- good_hab_list[[a]]
  accept_val <- length(hab_mets) # the value of the acceptable cells
  ## calculate total available habitat area
  sum_cells <- cellStats(b,stat='sum',na.rm=TRUE)/accept_val # divide by the
accept_val to get the total number of acceptable cells
  cell_size <- xres(b)^2
  tot_area <- cell_size*sum_cells

  area_look_tab <- data.frame(discharge = modeled_q,
                             total_area = tot_area) %>%
    dplyr::arrange(discharge)
  if(NormalizeByL == "Yes"){
    area_look_tab$normalized_area <- area_look_tab$total_area/reach_length #
normalize area by reach length
  }

  if(ExportTable == "Yes"){
    write.csv(area_look_tab, file = paste(reach_wd,
"dss_outputs/",reach_run,"_",fish, "_", a, "_aqllookup.csv", sep = ""))
  }

  return(area_look_tab)
} # end of function

```

## order.by.q.R

```

# Function: Puts rasters of total available habitat in order of ascending modeled
Q's
# Returns a list of rasters
# Last edited by Elaina Passero on 2/18/20

order.by.q <-
function(a,good_hab_list,modeled_q,ExportRast,reach_wd,fish,reach_run){
  if(nlayers(good_hab_list[[a]])==0){
    g <- brick(good_hab_list[[a]]) # ensures a single raster layer will "unstack"
  } else{
    g <- good_hab_list[[a]]
  }
  open_brick <- unstack(g)

```

```

names(open_brick) <- modeled_q
sort_q <- sort(modeled_q) # sorts from smallest to largest
rast_by_q <- list()
for(i in 1:length(sort_q)){ # builds new list with rasters in ascending order by
modeled Q
  n <- which(names(open_brick)==sort_q[i])
  rast_by_q[i] <- open_brick[n]
  if(ExportRast == "Yes"){
    writeRaster(open_brick[[n]],
                filename = paste(reach_wd, "dss_outputs/",reach_run,"_",fish,
                "_", sort_q[i], "_", a, "_", "hab.tif", sep = ""),
                format = "GTiff",overwrite=TRUE)
  }
} # end of for loop
names(rast_by_q) <- sort_q # names rasters in list by modeled Q
out_by_q <- list(rast_by_q, sort_q)

return(out_by_q)
}

```

## plot.a.q.curve.R

```

# This function will plot habitat area-discharge curves for all modeled flows
# Last edited by Elaina Passero on 6/11/20

plot.a.q.curve <-
function(fish_outputs,species_list,a,NormalizeByL,fish_tag,reach_run,native_list){

  # Gather area-lookup tables for lifestage
  tables <- lapply(species_list, function(species){
    fish_outputs[[species]]$area_look_tab[[a]]
  })
  names(tables) <- species_list

  all_a_tab <- bind_rows(tables,.id = "species")

  # plot in the same order as written in inputs
  all_a_tab$species <- ordered(all_a_tab$species)
  all_a_tab <- mutate(all_a_tab, plt_order = ifelse(species %in% native_list,1,2))

  if(NormalizeByL == "No"){
    plt <- ggplot(all_a_tab)+

```

```

geom_line(aes(x=discharge,y=total_area,color=reorder(species,plt_order)),size=1)+
  theme_few()+
  scale_colour_brewer(palette = "Set1",name="Species",labels=gsub("_"," ",
unique(all_a_tab$species)))+
  theme(text=element_text(size=14,face = "bold"),
        panel.border = element_rect(fill = NA,size=1),
        legend.position = c("bottom"),
        axis.text.x = element_text(colour = "black",face="plain"),
        axis.text.y = element_text(colour = "black",face="plain"))+
  labs(y=bquote('Total Habitat Area in '~m^2),x=bquote("Discharge in "~m^3/s))
} else{
  plt <- ggplot(all_a_tab)+

  geom_line(aes(x=discharge,y=normalized_area,color=reorder(species,plt_order)),size=1
)+
  theme_few()+
  scale_colour_brewer(palette = "Set1",name="Species",labels=gsub("_"," ",
unique(all_a_tab$species)))+
  theme(text=element_text(size=14,face = "bold"),
        panel.border = element_rect(fill = NA,size=1),
        legend.position = c("bottom"),
        axis.text.x = element_text(colour = "black",face="plain"),
        axis.text.y = element_text(colour = "black",face="plain"))+
  labs(y=bquote('Normalized Habitat Area in '~m^2/km),x=bquote("Discharge in
"~m^3/s))
}

ggsave(paste(reach_wd,"dss_outputs/figures/",reach_run,"_",fish_tag,"_a_q_curves_",
all.jpg",sep=""),
        plot=plt,width=8, height=5,units = "in",dpi = 300)
  return(plt)
}

chk <- gsub("_"," ",unique(all_a_tab$species))

```

## plot.a.q.curve.in.chan.R

```

# This function will plot habitat area-discharge curves for in channel flows
# Last edited by Elaina Passero on 6/12/20

```

```

plot.a.q.curve.in.chan <-
function(fish_outputs,species_list,a,NormalizeByL,fish_tag,reach_run,native_list){

  # Gather area-lookup tables for lifestage
  tables <- lapply(species_list, function(species){
    fish_outputs[[species]]$area_look_tab[[a]]
  })
  names(tables) <- species_list

  all_a_tab <- bind_rows(tables,.id = "species") %>%
    filter(discharge <= 8.5)

  # plot in the same order as written in inputs
  all_a_tab$species <- ordered(all_a_tab$species)
  all_a_tab <- mutate(all_a_tab, plt_order = ifelse(species %in% native_list,1,2))

  if(NormalizeByL == "No"){
    plt <- ggplot(all_a_tab)+

    geom_line(aes(x=discharge,y=total_area,color=reorder(species,plt_order)),size=1)+
      theme_few()+
      scale_colour_brewer(palette = "Set1",name="Species",labels=gsub("_"," ",
unique(all_a_tab$species)))+
      theme(text=element_text(size=14,face = "bold"),
            panel.border = element_rect(fill = NA,size=1),
            legend.position = c("bottom"),
            axis.text.x = element_text(colour = "black",face="plain"),
            axis.text.y = element_text(colour = "black",face="plain"))+
      labs(y=bquote('Total Habitat Area in '~m^2),x=bquote("Discharge in "~m^3/s))
  } else{
    plt <- ggplot(all_a_tab)+

    geom_line(aes(x=discharge,y=normalized_area,color=reorder(species,plt_order)),size=1
  )+
      theme_few()+
      scale_colour_brewer(palette = "Set1",name="Species",labels=gsub("_"," ",
unique(all_a_tab$species)))+
      theme(text=element_text(size=14,face = "bold"),
            panel.border = element_rect(fill = NA,size=1),
            legend.position = c("bottom"),
            axis.text.x = element_text(colour = "black",face="plain"),
            axis.text.y = element_text(colour = "black",face="plain"))+

```

```

      labs(y=bquote('Normalized Habitat Area in '~m^2/km),x=bquote("Discharge in
 '~m^3/s))
    }

  ggsave(paste(reach_wd,"dss_outputs/figures/",reach_run,"_",fish_tag,"_a_q_curves_",
  inchannel.jpg",sep=""),
        plot=plt,width=8, height=5,units = "in",dpi = 300)
  return(plt)
}

```

## post.processing.R (Module 3)

```

# This script will house the post-processing options and functionality
# Last edited by Elaina Passero on 6/9/20

# Load required packages
packages <-
c("SDMTools","sp","raster","rgeos","rgdal","sf","spatstat","spdep","tidyverse","raste
rVis",

"ggplot2","data.table","dplyr","spex","stars","igraph","deldir","hydroTSM",
  "lubridate","rlist")

# Check to see if each is installed, and install if not.
if (length(setdiff(packages, rownames(installed.packages())) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}

# load the installed libraries in the packages list
lapply(packages,library,character.only=TRUE)
#####

# User inputs
wd <- "C:/Users/epassero/Desktop/VRDSS/verde-refdss/" # Project working directory
setwd(wd)

hab_mets <- list("Depth","Velocity") #Variables from iRIC calculation result used for
habitat analysis - case sensitive
species_list <- c("Desert_Sucker","Sonora_Sucker","Roundtail_Chub","Speckled_Dace",
  "Smallmouth-Redeye_Bass","Red_Shiner") # species list
lifestages <- list("adult") #lifestages from oldest to youngest; must match order in
HSC table
veg_list <- c("drgt_tol_shrubs","tall_trees","popufre","prosvel","saligoo") #
vegetation group list

```

```

reach_name <- "USBeasley1" # Should match name of folder with results
model_run <- "2.12.20" # Should match end of name of folder with results
dis_unit <- "cms" #units of discharge
date_form <- "%m/%d/%Y" # format of date in flow scenarios. See ?as.Date for options.

# Yes or No. Indicate whether or not to substrate was a condition of suitable habitat
CheckSub <- "Yes"

# Yes or No. Choose whether or not to normalize habitat area by reach length
NormalizeByL <- "Yes"; if(NormalizeByL=="Yes"){
  reach_length <- 0.5
  length_unit <- "km"}

# Yes or No. Choose whether or not to calculate X-day statistics. Must supply number
of days.
CalcXDayStats <- "No"; if(CalcXDayStats=="Yes"){
  x_days <- 7} # number of days for moving area statistics

# Yes or No. Yes - limit analysis to supplied dates. No - consider entire hydrograph.
DateRange <- "No"; if(DateRange=="Yes"){
  start_date <- "1974-01-01" # "YYYY-MM-DD"
  end_date <- "1994-12-31"} # "YYYY-MM-DD"

# Flow scenarios to calculate metrics for
scene_names <- c("baseline_q","scenario_A","scenario_B")

#### End of User Inputs ####

#### Begin post-processing ####
reach_run <- paste(reach_name,"_",model_run,sep="")
reach_wd <- paste(wd,"reaches","/",reach_run,"/",sep = "")

# create list of flow scenario data frames
scene_list <- lapply(scene_names, function(s){
  q_ts <-
na.omit(fread(file=paste(reach_wd,"flow_scenarios","/",reach_name,"_",s,".csv",sep=""),
,
                                header=TRUE, sep = ",",data.table=FALSE)) %>%
  mutate(date_fix = as.Date(date,format = date_form)) %>%
  select(-date) %>%
  rename(date = date_fix)
  if(DateRange=="Yes"){
    q_ts <- subset(q_ts, date > as.Date(start_date))

```



```

    q_ts <- subset(q_ts, date < as.Date(end_date))
  }
  return(q_ts)
})
names(scene_list) <- scene_names

# Load pre.processing outputs
output_name <-
load(file=paste(reach_wd,"dss_outputs/","internal/",reach_run,"_pre_outputs.rdata",sep="")) # load rasterized 2D modeling results
eval(parse(text=paste("pre_outputs=",output_name)))

out_val_rast <- pre_outputs$out_val_rast
modeled_q <- pre_outputs$modeled_q

# Fish post-processing
all_fish_post_out <- list()
all_fish_post_out <- lapply(scene_names, function(scene_name){

  s <- scene_list[[scene_name]]

  post_fish_outputs <- list()
  post_fish_outputs <- lapply(species_list, function(species){

    output_name <-
load(file=paste(reach_wd,"dss_outputs/","internal/",reach_run,"_",species,"_fish_outputs.rdata",sep="")) # load fish outputs
eval(parse(text=paste("one_spec=",output_name)))

    for(i in 1:length(one_spec)){ # extracts the outputs by species into their own
object
      tempobj = one_spec[[i]]
      eval(parse(text=paste(names(one_spec)[[i]],"=tempobj")))
    }

    # Flow-Scenario related scripts
    source("interp.table.R")
    inter_tab <- lapply(lifestages, function(a)
interp.table(a,s,area_look_tab,NormalizeByL))
    names(inter_tab) <- lifestages
    post_fish_outputs$inter_tab <- inter_tab

```

```

# Generate and view plots of total area through the hydrograph
source("interp.plot.R")
inter_plots <- interp.plot(inter_tab, NormalizeByL, species, scene_name, reach_run)
names(inter_plots) <- species
post_fish_outputs$inter_plots <- inter_plots

## Generate Data Frames of moving X-Day area and discharge statistics
if(CalcXDayStats=="Yes"){
  source("x.day.stats.R")
  x_day_stats <- lapply(lifestages, function(a)
x.day.stats(a,inter_tab,x_days,NormalizeByL,reach_length))
  names(x_day_stats) <- lifestages
  post_fish_outputs$x_day_stats <- x_day_stats
}

source("avg.month.area.R")
avg_monthly_area <- lapply(lifestages, function(a)
avg.month.area(a,inter_tab,NormalizeByL))
names(avg_monthly_area) <- lifestages
post_fish_outputs$avg_monthly_area <- avg_monthly_area
# end of flow scenario dependent processes

# save outputs (dss_outputs) for use in scenario.comparison.R

list.save(post_fish_outputs, file=paste(reach_wd, "dss_outputs/", "internal/", reach_run,
"_", scene_name, "_post_", species, ".rdata", sep=""))

return(post_fish_outputs)
}) # end of single species function

names(post_fish_outputs) <- species_list

return(post_fish_outputs)
}) # end of flow scenario list function
names(all_fish_post_out) <- names(scene_list)

# Riparian vegetation post-processing
all_veg_post_out <- list()
all_veg_post_out <- lapply(scene_names, function(scene_name){

  s <- scene_list[[scene_name]]

  source("q.ep.weibull.R")

```

```

hydro_ep <- q.ep.weibull(s,modeled_q,scene_name)
all_veg_post_out$hydro_ep <- hydro_ep

source("make.inun.q.map.R")
wet_map <- make.inun.q.map(hydro_ep,out_val_rast,modeled_q,scene_name,reach_run)
all_veg_post_out$wet_map <- wet_map

source("make.ep.maps.R")
ep_map <- make.ep.maps(hydro_ep,out_val_rast,modeled_q,scene_name)
all_veg_post_out$ep_map <- ep_map

# make maps of probability of occurrence of vegetation
all_veg_logit <-
fread(paste(reach_wd,"habitat_info","/",reach_name,"_veg_pref",".csv",sep=""),
      header=TRUE, sep = ",",data.table = FALSE) # load logistic
equations

source("find.veg.logit.R"); source("find.prob.occure.R"); source("make.veg.maps.R")
prob_veg_maps <- lapply(veg_list, function(v) {
  one_veg_logit <- find.veg.logit(all_veg_logit,v)
  hydro_ep_prob <- find.prob.occure(v,hydro_ep,one_veg_logit) # outputs table of
discharge, EP of discharge, and Prob of Veg for discharge
  prob_veg_map <- make.veg.maps(v,ep_map,hydro_ep_prob,scene_name)

  save(prob_veg_map,
file=paste(reach_wd,"dss_outputs/", "internal/",reach_run,"_",scene_name,"_post_",v,".
RData",sep=""))

  return(prob_veg_map)
})
names(prob_veg_maps) <- veg_list
all_veg_post_out$prob_veg_maps <- prob_veg_maps

return(all_veg_post_out)
}) # end of flow scenario list function
names(all_veg_post_out) <- names(scene_list)

```

## interp.table.R

```

# This function will add total available and total normalized area (if applicable)
to the hydrograph table

```

```

# Will use simple linear interpolation to fill in the areas for any discharges not
run in iRIC.
# Last updated by Elaina Passero on 12/4/19

interp.table <- function(a,hydrograph,area_look_tab,NormalizeByL){
  a_l_t <- area_look_tab[[a]]
  unique_q <- a_l_t %>%
    bind_rows(data.frame(anti_join(hydrograph["discharge"],a_l_t["discharge"])))
  %>% # all discharges that will be in area-lookup table
    unique(.)

  if(NormalizeByL=="Yes"){
    unique_q <- unique_q %>%
      bind_rows(data.frame(discharge=0,total_area=0,normalized_area=0)) %>% #
temporary measure to address historic flows below minimum modeled Q
      dplyr::arrange(discharge)

    int_tot <- approx(unique_q,method="linear",xout=unique_q$discharge) # linearly
interpolate for missing area values

    int_norm_len <- unique_q %>%
      select(-total_area) %>%
      approx(.,method="linear",xout=unique_q$discharge) # linearly interpolate for
missing normalized area values
    int_uni_q <- data.frame(int_tot,int_norm_len[[2]])
    colnames(int_uni_q) <- c("discharge","total_area","normalized_area")
  } else{
    unique_q <- unique_q %>%
      bind_rows(data.frame(discharge=0,total_area=0)) %>% # temporary measure to
address historic flows below minimum modeled Q
      dplyr::arrange(discharge)
    int_tot <- approx(unique_q,method="linear",xout=unique_q$discharge) # linearly
interpolate for missing area values
    int_uni_q <- data.frame(int_tot)
    colnames(int_uni_q) <- c("discharge","total_area")
  }

  hydrograph$total_area <-
int_uni_q[match(hydrograph$discharge,int_uni_q$discharge),"total_area"]
  if(NormalizeByL=="Yes"){
    hydrograph$normalized_area <-
int_uni_q[match(hydrograph$discharge,int_uni_q$discharge),"normalized_area"]
  }
}

```

```

    return(hydrograph)
  }

```

## interp.plot.R

```

# The function generates plots of total available and normalized habitat area and
# discharge through the hydrograph
# Last updated by Elaina Passero on 2/18/20
# scales need to be fixed on plots

interp.plot <- function(inter_tab, NormalizeByL, species, scene_name, reach_run){

  # one data frame for all lifestages
  if(length(lifestages) > 1){
    i_tab_all_age <- bind_rows(inter_tab, .id = "lifestage")
  } else{
    i_tab_all_age <- inter_tab[[1]] %>%
      mutate(lifestage = lifestages[[1]])
  }

  if(NormalizeByL=="Yes"){
    # create longer df
    i_tab_all_age <- i_tab_all_age %>%
      pivot_longer(c(discharge, normalized_area), names_to = "metric")

    met.labs <- c("Discharge", "Normalized Habitat Area")
    names(met.labs) <- c("discharge", "normalized_area")
    my_lims <- function(y) {if (max(y) <= max(modeled_q+500)) c(0, max(modeled_q))
      else c(0, max(i_tab_all_age$value, na.rm = TRUE))}

    p_norm <- ggplot(i_tab_all_age)+
      geom_line(data =
filter(i_tab_all_age, metric=="discharge"), aes(x=date, y=value))+
      geom_line(data =
filter(i_tab_all_age, metric=="normalized_area"), aes(x=date, y=value, linetype=lifestage
)))+
      theme_gray()+
      theme(text=element_text(size=14, face = "bold", color = "black"),
        panel.border = element_rect(fill = NA, size=1),
        strip.background = element_rect(fill = NA),
        axis.text.x = element_text(colour = "black", face="plain"),
        axis.text.y = element_text(colour = "black", face="plain"))+

```

```

    facet_wrap(~metric)+
    facet_grid(rows = vars(metric),labeller = labeller(metric = met.labs),scales =
"free_y")+
    scale_y_continuous(limits=my_lims)+
    labs(x="Date",y=NULL,title = paste(scene_name,species,sep=" - "))

plt <- list(p_norm)
} else{
  # create longer df
  i_tab_all_age <- i_tab_all_age %>%
    pivot_longer(c(discharge,total_area),names_to = "metric")

  met.labs <- c("Discharge","Total Habitat Area")
  names(met.labs) <- c("discharge","total_area")
  my_lims <- function(y) {if (max(y) <= max(modeled_q)) c(0,max(modeled_q))
    else c(0,max(i_tab_all_age$value,na.rm = TRUE))}

  p_tot <- ggplot(i_tab_all_age)+
    geom_line(data =
filter(i_tab_all_age,metric=="discharge"),aes(x=date,y=value),size=1)+
    geom_line(data =
filter(i_tab_all_age,metric=="total_area"),aes(x=date,y=value,linetype=lifestage))+
    theme_gray()+
    theme(text=element_text(size=14,face = "bold",color = "black"),
          panel.border = element_rect(fill = NA,size=1),
          strip.background = element_rect(fill = NA),
          axis.text.x = element_text(colour = "black",face="plain"),
          axis.text.y = element_text(colour = "black",face="plain"))+
    facet_wrap(~metric)+
    facet_grid(rows = vars(metric),labeller = labeller(metric = met.labs),scales =
"free_y")+
    scale_y_continuous(limits=my_lims)+
    labs(x="Date",y=NULL,title = paste(scene_name,species,sep=" - "))

  plt <- list(p_tot)
}

ggsave(paste(reach_wd,"dss_outputs/figures/",reach_run,"_",scene_name,"_",species,"_
", "full.png",sep=""),plot=plt[[1]],width=7, height=5,units = "in")
return(plt)
}

```

## x.day.stats.R

```
# This function calculates trailing X-day minimum, maximum, and mean areas
# Returns a new data frame containing the metrics by date
# Last edited by Elaina Passero 01/22/20

x.day.stats <- function(a,inter_tab,x_days,NormalizeByL,reach_length){
  if(NormalizeByL == "Yes"){
    move_df <- data.frame(date=inter_tab[[a]]$date,
                          actual_q = inter_tab[[a]]$discharge,
                          actual_tot_area = inter_tab[[a]]$total_area,
                          actual_norm_area = inter_tab[[a]]$normalized_area,
                          x_day_avg_area = NA, x_day_avg_norm_area = NA,
                          x_day_min_area = NA, x_day_min_norm_area = NA,
                          x_day_max_area = NA, x_day_max_norm_area = NA)

    all_days <- length(move_df$date)
    for(i in 1:(all_days-x_days)){
      last_day <- x_days+i-1
      run_days <- dplyr::slice(inter_tab[[a]],i:last_day) # creates subset of
observations for X-days
      move_df$x_day_avg_area[last_day] <- mean(run_days$total_area)
      move_df$x_day_avg_norm_area[last_day] <- mean(run_days$normalized_area)
      move_df$x_day_min_area[last_day]<-min(run_days$total_area)
      move_df$x_day_min_norm_area[last_day]<-min(run_days$normalized_area)
      move_df$x_day_max_area[last_day]<-max(run_days$total_area)
      move_df$x_day_max_norm_area[last_day]<-max(run_days$normalized_area)
    }
  } else {
    move_df <- data.frame(date=inter_tab[[a]]$date,
                          actual_q = inter_tab[[a]]$discharge,
                          actual_area = inter_tab[[a]]$total_area,
                          x_day_avg_area = NA,
                          x_day_min_area = NA,
                          x_day_max_area = NA)

    all_days <- length(move_df$date)
    for(i in 1:(all_days-x_days)){
      last_day <- x_days+i
      run_days <- dplyr::slice(inter_tab[[a]],i:last_day) # creates subset of
observations for X-days
      move_df$x_day_avg_area[last_day] <- mean(run_days$total_area)
      move_df$x_day_min_area[last_day]<-min(run_days$total_area)
      move_df$x_day_max_area[last_day]<-max(run_days$total_area)
    }
  }
}
```

```

    }

    return(move_df)
  }

```

## avg.month.area.R

```

# This function calculates monthly area using central 10-day lowest consecutive
# area values
# Returns a data frame with average monthly area
# should update to run without normalizing area
# Last edited by Elaina Passero on 11/21/19

avg.month.area <- function(a,inter_tab,NormalizeByL){
  # Monthly stats using lowest consecutive 10 days
  move_df <- data.frame(date=inter_tab[[a]]$date,low10d_avg_area=NA)
  all_days <- length(move_df$date)
  for(i in seq(5,(all_days-5),1)){
    day_1 <- i-4
    day_last <- i+5
    run_days <- dplyr::slice(inter_tab[[a]],day_1:day_last) # creates subset of
    observations for X-days
    if(NormalizeByL == "Yes"){
      move_df[i,2] <- mean(run_days$normalized_area)
    } else {
      move_df[i,2] <- mean(run_days$total_area)
    }
  }
  move_df$day <- as.numeric(format(as.Date(move_df$date,format="%m/%d/%Y"),"%d"))
  low10df <- filter(move_df, is.na(move_df$low10d_avg_area)==FALSE)
  runMonth <- as_tibble(low10df) %>%
    group_by(year(date)) %>%
    group_by(month(date),add=TRUE) %>%
    summarise(mon_avg_low_area=min(low10d_avg_area)) %>%
    rename(year="year(date)",month="month(date)")
  return(runMonth)
}

```

## interp.plot.abbr.R

```

# The function generates plots of total available and normalized habitat area and
# discharge through the hydrograph

```



```

# This time series is for a subset of time
# Last updated by Elaina Passero on 12/05/19

interp.plot.abbr <- function(inter_tab, NormalizeByL, day1, day2, scenario, species){

  # one data frame for all lifestages
  if(length(lifestages) > 1){
    i_tab_all_age <- bind_rows(inter_tab, .id = "lifestage") %>%
      pivot_longer()
  } else{
    i_tab_all_age <- inter_tab[[1]] %>%
      mutate(lifestage = lifestages[[1]])
  }

  if(NormalizeByL=="Yes"){
    # create longer df
    i_tab_all_age <- i_tab_all_age %>%
      pivot_longer(c(discharge, normalized_area), names_to = "metric") %>%
      filter(date >= day1 & date <= day2)

    met.labs <- c("Discharge", "Normalized Habitat Area")
    names(met.labs) <- c("discharge", "normalized_area")
    my_lims <- function(y) {if (max(y) <= 200) c(0, 200) # summer limit
      else c(0, max(i_tab_all_age$value, na.rm = TRUE))}

    p_norm <- ggplot(i_tab_all_age)+
      geom_line(data =
filter(i_tab_all_age, metric=="discharge"), aes(x=date, y=value))+
      geom_line(data =
filter(i_tab_all_age, metric=="normalized_area"), aes(x=date, y=value, linetype=lifestag
e))+
      theme_gray()+
      theme(text=element_text(size=14, face = "bold", color = "black"),
        panel.border = element_rect(fill = NA, size=1),
        strip.background = element_rect(fill = NA),
        axis.text.x = element_text(colour = "black", face="plain"),
        axis.text.y = element_text(colour = "black", face="plain"))+
      facet_wrap(~metric)+
      facet_grid(rows = vars(metric), labeller = labeller(metric = met.labs), scales =
"free_y")+
      scale_y_continuous(limits=my_lims)+
      labs(x="Date", y=NULL, title = paste(scenario, " - ", species, " ", day1, " to
", day2, sep=""))
  }
}

```

```

plt <- list(p_norm)
} else{
  # create longer df
  i_tab_all_age <- i_tab_all_age %>%
    pivot_longer(c(discharge,total_area),names_to = "metric") %>%
    filter(date >= day1 & date <= day2)

  met.labs <- c("Discharge","Total Habitat Area")
  names(met.labs) <- c("discharge","total_area")
  my_lims <- function(y) {if (max(y) <= max(modeled_q)) c(0,max(modeled_q))
    else c(0,max(i_tab_all_age$value,na.rm = TRUE))}

  p_tot <- ggplot(i_tab_all_age)+
    geom_line(data =
filter(i_tab_all_age,metric=="discharge"),aes(x=date,y=value),size=1)+
    geom_line(data =
filter(i_tab_all_age,metric=="total_area"),aes(x=date,y=value,linetype=lifestage))+
    theme_gray()+
    theme(text=element_text(size=14,face = "bold",color = "black"),
      panel.border = element_rect(fill = NA,size=1),
      strip.background = element_rect(fill = NA),
      axis.text.x = element_text(colour = "black",face="plain"),
      axis.text.y = element_text(colour = "black",face="plain"))+
    facet_wrap(~metric)+
    facet_grid(rows = vars(metric),labeller = labeller(metric = met.labs),scales =
"free_y")+
    scale_y_continuous(limits=my_lims)+
    labs(x="Date",y=NULL,title = paste(scenario," - ",species," ",day1," to
",day2,sep=""))

  plt <- list(p_tot)
}

return(plt)
}

```

## q.ep.weibull.R

```

# this function will calculate EP for flows in flow scenario and those above it
# If flows above the range of the flow scenario are input, EP will be calculated
from Weibull plotting position assuming 1 event per Q

```

```

# Last edited by Elaina Passero 12/09/19

q.ep.weibull <- function(hydrograph, modeled_q, scene_name){

  # Calculate exceedence probability for flows in scenario
  flow_scene_ep <- data.frame(discharge = hydrograph["discharge"], q_rank = rank(-
hydrograph["discharge"], ties.method = "min")) # ranks discharges; same values get
same ranks
  n <- as.numeric(length(flow_scene_ep$discharge))
  flow_scene_ep$EP <- flow_scene_ep$q_rank / (1+n)

  df_m_q <- data.frame(discharge = modeled_q) %>%
    filter(discharge < max(hydrograph$discharge)) # modeled discharges within range
of flow scenario

  hydro_ep <-
bind_rows(flow_scene_ep, data.frame(anti_join(df_m_q["discharge"], flow_scene_ep["disc
harge"]))) %>%
  arrange(discharge) %>%
  distinct() # removes duplicate entries (does not affect EP values)

  hydro_ep <-
data.frame(approx(x = hydro_ep$discharge, y = hydro_ep$EP, method = "linear", xout = hydro_ep$d
ischarge)) %>%
  dplyr::rename(discharge = x, EP = y)

  # sets EP for 0 to 1 if 0 was included in the analysis
  if(min(hydro_ep$discharge) == 0){
    hydro_ep[1,2] <- 1
  }

  # Calculate EP for flows smaller than flow scenario.
  if(any(modeled_q < min(hydrograph$discharge))){
    df_m_q_smaller <- data.frame(discharge = modeled_q) %>%
      filter(discharge < min(hydrograph$discharge)) %>% # modeled discharges below
range of flow scenario
      mutate(EP = 1) # sets EP

    hydro_ep <- hydro_ep %>%
      filter(discharge > min(hydrograph$discharge)) %>%
      bind_rows(df_m_q_smaller) %>% # join extra flows and their EP values to flow
scenario
    dplyr::arrange(discharge)
  }
}

```

```

}

# Calculalte EP for flows larger than flow scenario. Assuming 1 event per flow.
if(isTRUE(modeled_q > max(hydrograph$discharge))){
  df_m_q_larger <- data.frame(discharge=modeled_q) %>%
    filter(discharge > max(hydrograph$discharge)) %>% # modeled discharges above
range of flow scenario
  mutate(q_rank = rank(-df_m_q_larger$discharge,ties.method = "min")) %>%
  mutate(EP = q_rank/(1+n+length(df_m_q_larger)))

  hydro_ep <- bind_rows(hydro_ep,df_m_q_larger) %>% # join extra flows and their
EP values to flow scenario
  select(-q_rank)
}

#write.csv(hydro_ep,file=paste(reach_wd,reach_name,"_",scene_name,"_ep_table.csv",se
p=""))

return(hydro_ep)
}

```

## make.inun.q.map.R

```

# this function will make inundating Q maps for presentation
# Last edited by Elaina Passero on 1/24/20

make.inun.q.map <- function(hydro_ep,out_val_rast,modeled_q,scene_name,reach_run){

  source("order.by.q.R")
  hold <- order.by.q(1,out_val_rast,modeled_q,"No")
  brick_by_q <- brick(hold[[1]])
  sort_q <- hold[[2]]

  rclmat <- matrix(c(0,Inf,1, -Inf,0,NA),ncol=3,byrow=TRUE)
  wet_brick <- reclassify(brick_by_q,rclmat)
  for(j in 1:nlayers(wet_brick)){

```

```

        wet_brick[[j]] <- wet_brick[[j]]*sort_q[j] # apply discharge value to inundated
areas
    }

    # use cover() function to make layer of inundating discharge
    wet_map <- wet_brick[[1]]

    for(j in 1:nlayers(wet_brick)){
        wet_map <- cover(wet_map,wet_brick[[j]]) # inundating Q map
    }

    writeRaster(wet_map,filename =
paste(reach_wd,"dss_outputs/",reach_run,"_",scene_name,"_wet_map.tif",sep=""),
        format="GTiff",overwrite = TRUE)
    return(wet_map)
}

```

## make.ep.maps.R

```

# this function will reclassify inundating q map with EP values from a flow
scenario.
# last edited by Elaina Passero on 12/09/19

make.ep.maps <- function(hydro_ep,out_val_rast,modeled_q,scene_name){

    source("order.by.q.R")
    hold <- order.by.q(1,out_val_rast,modeled_q,"No")
    brick_by_q <- brick(hold[[1]])
    sort_q <- hold[[2]]

    rclmat <- matrix(c(0,Inf,1, -Inf,0,NA),ncol=3,byrow=TRUE)
    wet_brick <- reclassify(brick_by_q,rclmat)
    for(j in 1:nlayers(wet_brick)){
        wet_brick[[j]] <- wet_brick[[j]]*sort_q[j] # apply discharge value to inundated
areas
    }

    # use cover() function to make layer of inundating discharge
    wet_map <- wet_brick[[1]]

    for(j in 1:nlayers(wet_brick)){
        wet_map <- cover(wet_map,wet_brick[[j]]) # inundating Q map
    }
}

```

```

}

# get EP values for modeled discharges
modeled_q_ep <- hydro_ep %>%
  filter(discharge %in% modeled_q) %>%
  distinct() %>%
  arrange(discharge)

# build matrix for reclassification
rcl_ep <- data.frame(from = modeled_q_ep$discharge,
  to = c(modeled_q_ep$discharge[-1], Inf),
  becomes = modeled_q_ep$EP) %>%
  as.matrix(.)

# reclassify raster
ep_map <- reclassify(wet_map, rcl_ep, right = FALSE)

writeRaster(ep_map, filename =
paste(reach_wd, "dss_outputs/", reach_name, "_", scene_name, "_ep_map.tif", sep=""),
  format="GTiff", overwrite = TRUE)
return(ep_map)
}

```

## find.veg.logit.R

```

# This function pulls riparian vegetation logistic regression equations for a
single group
# Last edited by Elaina Passero on 11/20/19

find.veg.logit <- function(all_veg_logit,v){
  col_num <- grep(v,names(all_veg_logit),ignore.case = TRUE)
  titles <- names(all_veg_logit)
  one_veg_logit <- all_veg_logit[col_num]
  return(one_veg_logit)
}

```

## find.prob.occur.R

```

# This function will calculate probability of occurrence of vegetation
guilds/species from EP
# Last edited by Elaina Passero on 11/20/19

```

```

find.prob.occure <- function(v,hydro_ep,one_veg_logit){

  nums <- seq(1:(length(one_veg_logit[[1]])-1))
  vars <- paste("B", (nums-1), sep="")
  for(i in 1:length(nums)){
    assign(vars[i],as.numeric(one_veg_logit[i+1,]))
  }

  # create list of variables and their values
  var_vals <- list()
  for(i in 1:length(nums)){
    var_vals[[i]] <- as.numeric(one_veg_logit[i+1,]) # each variable becomes its
own object
  }
  names(var_vals) <- vars

  eqn <- one_veg_logit[1,1] # extract equation from table
  for(i in 1:length(vars)){eqn <- gsub(vars[i],var_vals[i],eqn)} # add coefficients
to eqn

  # evaluate the equation at each EP value
  for(i in 1:length(hydro_ep$EP)){
    x <- hydro_ep$EP[i]
    eqn_x <- gsub("*x",paste("*",x,sep=""),eqn,fixed=TRUE) # replaces x with EP
value
    hydro_ep$prob_v[i] <- eval(parse(text=eqn_x))
  }
  return(hydro_ep)
}

```

## make.veg.maps.R

```

# This script will make maps of probability of occurrence of vegetation
guilds/species
# Last edited by Elaina Passero on 01/24/20

make.veg.maps <- function(v,ep_map,hydro_ep_prob,scene_name){

  # build matrix for reclassification
  rcl_ep <- data.frame(from = c(hydro_ep_prob$EP[-1],0),
                      to = hydro_ep_prob$EP,
                      becomes = hydro_ep_prob$prob_v) %>%

```

```

as.matrix(.)

# reclassify raster
prob_veg_map <- reclassify(ep_map, rcl_ep, right = TRUE)

writeRaster(prob_veg_map,filename =
paste(reach_wd,"dss_outputs/",reach_name,"_",scene_name,"_",v,"_prob_map.tif",sep=""
),
          format="GTiff",overwrite=TRUE)
return(prob_veg_map)
}

```

## scenario.comparison.Rmd (Module 4)

```

---
title: "scenario_eval"
author: "Elaina Passero"
date: "June 9, 2020"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r packages, message=FALSE, warning=FALSE, include=FALSE, paged.print=FALSE}
# Load required packages
packages <-
c("SDMTools","sp","raster","rgeos","rgdal","sf","spatstat","spdep","tidyverse","rast
erVis",

"ggplot2","gridExtra","data.table","dplyr","spex","stars","igraph","deldir","hydroTS
M",

      "lubridate","rlist","ggthemes","ggpubr")
# Check to see if each is installed, and install if not.
if (length(setdiff(packages, rownames(installed.packages())) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}
# load the installed libraries in the packages list
lapply(packages,library,character.only=TRUE)
```

```



## ## User Inputs

```
*Instructions for defining user inputs.
```{r user_inputs}
# User inputs
post_tag <- "fin2" # unique code to identify current run of DSS
wd <- "C:/Users/epassero/Desktop/VRDSS/verde-refdss/" # Project working directory
setwd(wd)
hab_mets <- list("Depth", "Velocity") #Variables from iRIC calculation result used
for habitat analysis - case sensitive
species_list <- c("Desert_Sucker", "Sonora_Sucker", "Roundtail_Chub", "Speckled_Dace",
                  "Smallmouth-Redeye_Bass", "Red_Shiner") # species list
native_list <- c("Desert_Sucker", "Sonora_Sucker", "Roundtail_Chub", "Speckled_Dace") #
list of native fish species in species_list
lifestages <- list("adult") #lifestages from oldest to youngest; must match order in
HSC table
veg_list <- c("drgt_tol_shrubs", "tall_trees", "popufre", "prosvet", "saligoo") #
vegetation group list
reach_name <- "USBeasley1" # Should match name of folder with results
model_run <- "2.12.20" # Should match end of name of folder with results
dis_unit <- "cms" #units of discharge
date_form <- "%m/%d/%Y" # format of date in flow scenarios. See ?as.Date for
options.
# Flow scenarios to include. This list should start with "baseline_q"
scene_names <- c("baseline_q", "scenario_A", "scenario_B")
# Yes or No. Choose whether or not to normalize habitat area by reach length
NormalizeByL <- "Yes"; if(NormalizeByL=="Yes"){
  reach_length <- 0.5
  length_unit <- "km"}
# Yes or No. Yes - Limit analysis to supplied dates. No - consider entire
hydrograph.
DateRange <- "No"; if(DateRange=="Yes"){
  start_date <- "1974-01-01" # "YYYY-MM-DD"
  end_date <- "1994-12-31" # "YYYY-MM-DD"
  ...
}
```{r scene_load, echo=FALSE}
reach_run <- paste(reach_name, "_", model_run, sep="")
reach_wd <- paste(wd, "reaches", "/", reach_run, "/", sep = "")
# create list of flow scenario data frames
scene_list <- lapply(scene_names, function(s){
```

```

q_ts <-
na.omit(fread(file=paste(reach_wd, "flow_scenarios", "/", reach_name, "_", s, ".csv", sep="
"),
                                header=TRUE, sep = ",", data.table=FALSE)) %>%
mutate(date_fix = as.Date(date, format = date_form)) %>%
select(-date) %>%
rename(date = date_fix)
if(DateRange=="Yes"){
  q_ts <- subset(q_ts, date > as.Date(start_date))
  q_ts <- subset(q_ts, date < as.Date(end_date))
}
return(q_ts)
})
names(scene_list) <- scene_names
names(species_list) <- species_list
...

## Generating Figures and Tables for Fish
```{r fish_figs_tabs, echo=FALSE, message=FALSE, warning=FALSE}
# Figures and tables for fish
post_fish_figs <- list()
post_fish_figs <- lapply(species_list, function(species){

  # Load all scenario results for species
  scene_fish_out <- list()
  scene_fish_out <- lapply(scene_names, function(scene_name){
    output_name <- load(file=paste(reach_wd, "dss_outputs/", "internal/",

reach_run, "_", scene_name, "_post_", species, ".rdata", sep="")) # Load post-fish outputs
    eval(parse(text=paste("hold_fish_output=", output_name)))
    return(hold_fish_output)
  })
  names(scene_fish_out) <- scene_names

  source("plot.10day.ts.R")
  ten_day_ts_plt <- lapply(lifestages, function(a)
plot.10day.ts(a, species, scene_fish_out, scene_list, NormalizeByL, post_tag))
  names(ten_day_ts_plt) <- lifestages
  post_fish_figs$ten_day_ts_plt <- ten_day_ts_plt

  source("build.10day.tables.R")
  ten_day_min_outputs <- lapply(lifestages, function(a)
build.10day.tables(a, species, scene_list, scene_fish_out, post_tag))
  names(ten_day_min_outputs) <- lifestages

```

```

post_fish_figs$ten_day_min_outputs <- ten_day_min_outputs

source("make.barplots.chg.R")
chg_barplots <- lapply(lifestages, function(a)
make.barplots.chg(a,species,ten_day_min_outputs,post_tag,scene_names_full))
names(chg_barplots) <- lifestages
post_fish_figs$chg_barplots <- chg_barplots

return(post_fish_figs)
})
names(post_fish_figs) <- species_list
# pull % change
fish_pc_builder <- lapply(species_list, function(species){

  one_spec_list <- lapply(lifestages, function(a){ # pull % change tables for all
lifestages from a single species
    post_fish_figs[[species]]$ten_day_min_outputs[[a]]$sum_all_yrs_dif
  })
  names(one_spec_list) <- lifestages
  one_spec_df <- bind_rows(one_spec_list,.id = "lifestage") # build single df per
species
  return(one_spec_df)
})
names(fish_pc_builder) <- species_list
# add criteria - nativeness
fish_pc_tab <- bind_rows(fish_pc_builder,.id = "species") %>%
  filter(scene != "baseline_q") %>%
  mutate(criteria = ifelse(species %in% native_list,"native","non-native"))
...

### Fish Figures
```{r fish_figs, echo=FALSE, message=FALSE,
warning=FALSE,results='hide',fig.keep='all'}
# Load barplots of % change
plot.bars <- function(one_fish_metrics,title){
  one_fish_metrics$chg_barplots
}
map2(post_fish_figs,species_list,plot.bars)
fish_pc_tab <- fish_pc_tab %>%
  mutate(plt_order = ifelse(species %in% native_list,1,2))

#non-natives plotted at the end
fish_pc_tab$species <- as.factor(fish_pc_tab$species)
# plots of % change by species (all lifestages) faceted by scenario

```

```

chg_f_plt <- ggplot(data = fish_pc_tab, aes(x = factor(month), y = per_chg,
      fill = reorder(species,plt_order), alpha =
lifestage))+
  theme_few()+
  scale_fill_brewer(palette = "Set1",labels=fish_labels)+
  geom_bar(stat = "identity",position = "dodge")+
  theme(legend.position = "bottom",
    text=element_text(size=14,face = "bold",color = "black"),
    axis.text.x = element_text(colour = "black",face="plain"),
    axis.text.y = element_text(colour = "black",face="plain"))+
  facet_wrap(vars(scene),dir = "v")+
  scale_x_discrete(breaks=c(1,2,3,4,5,6,7,8,9,10,11,12),
    labels =
c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"))+
  {if(length(lifestages)>1) scale_alpha_discrete(range = c(1,0.35))}+
  {if(length(lifestages)==1) scale_alpha_discrete(guide="none",range = c(1,1))}+
  labs(fill = "Species")+
  ylab("% Change in Habitat Area")+
  xlab("Month")

ggsave(paste(reach_wd,"dss_outputs/figures/",reach_run,"_",post_tag,"_", "all_fish_ch
g.png",sep=""),
      plot=chg_f_plt,width=8, height=5,units = "in")
...

## Generating Figures and Tables for Vegetation
``{r veg_figs_tabs, echo=FALSE, message=FALSE, warning=FALSE}
# Load inundating discharge map
wet_map <-
raster(paste(reach_wd,"dss_outputs/",reach_run,"_baseline_q_wet_map.tif",sep=""))
sum_veg_metrics <- list()
sum_veg_metrics <- lapply(veg_list, function(v) {

  scene_veg_out <- lapply(scene_names, function(scene_name){
    output_name <-
load(file=paste(reach_wd,"dss_outputs/","internal/",reach_run,"_",scene_name,"_post_
",v,".RData",sep=""))
    eval(parse(text=paste("hold_veg_output=",output_name)))
    return(hold_veg_output)
  })
  names(scene_veg_out) <- scene_names

# identify areas of high probability of occurrence
source("id.high.prob.areas.R")

```

```

    high_prob_areas <- lapply(scene_names, function(scene_name)
id.high.prob.areas(scene_veg_out, scene_name))
    names(high_prob_areas) <- scene_names
    sum_veg_metrics$high_prob_areas <- high_prob_areas

    # maps of change in high probability of occurrence areas
    source("map.chg.veg.areas.R")
    chg_high_prob_areas <- lapply(scene_names, function(scene_name)
map.chg.veg.areas(high_prob_areas, scene_name))
    names(chg_high_prob_areas) <- scene_names
    sum_veg_metrics$chg_high_prob_areas <- chg_high_prob_areas

    # Tabulate area of high probability of occurrence
    source("make.hp.area.tables.R")
    hp_area_tab <-
make.hp.area.tables(v, high_prob_areas, scene_names, NormalizeByL, reach_Length)
    sum_veg_metrics$hp_area_tab <- hp_area_tab

    # Calculate % change in high probability of occurrence areas
    source("calc.chg.veg.area.R")
    per_chg_hp_tab <- calc.chg.veg.area(hp_area_tab, scene_names, NormalizeByL)
    sum_veg_metrics$per_chg_hp_tab <- per_chg_hp_tab

    # describe movement of high probability of occurrence areas
    source("check.veg.movement.R")
    veg_movement <- lapply(scene_names, function(scene_name)
check.veg.movement(wet_map, high_prob_areas, scene_name))
    names(veg_movement) <- scene_names
    sum_veg_metrics$veg_movement <- veg_movement

    return(sum_veg_metrics)
  })
  names(sum_veg_metrics) <- veg_list
  ...
### Vegetation Figures
  ```{r veg_figs, echo = FALSE, results='hide', fig.keep='all'}```
  # Show maps of high probability of occurrence areas
  plot.rast <- function(one_veg_metrics, title){
    one_veg_stack <- stack(one_veg_metrics$high_prob_areas) # stacks all scenarios for
a single guild
    lpt <- levelplot(one_veg_stack, margin = FALSE,
                      region = TRUE,
                      scales=list(draw=FALSE),

```

```

        at = c(-1,.5,1.5),
        col.regions = c("grey","black"),
        colorkey = FALSE)
    return(lpt)
}
map2(sum_veg_metrics,veg_list,plot.rast)
...

Maps of high probability of occurrence areas (black).
## Combination Figures
```{r combo_fig_veg, echo=FALSE}
#### Build Vegetation Tables
# pull % change
veg_pc_builder <- lapply(veg_list, function(v){
  per_chg_list <- sum_veg_metrics[[v]]$per_chg_hp_tab
})
names(veg_pc_builder) <- veg_list
veg_pc_tab <- bind_rows(veg_pc_builder,.id = "group")
# pull criteria - movement
veg_m_builder <- lapply(veg_list, function(v){
  move_tab <- data.frame(sum_veg_metrics[[v]]$veg_movement) %>%
    select(-baseline_q) %>%
    pivot_longer(everything(),names_to = "scene",values_to = "criteria")
})
names(veg_m_builder) <- veg_list
veg_m_tab <- bind_rows(veg_m_builder,.id = "group")
# add criteria to % change table
veg_summary_tab <- inner_join(veg_pc_tab,veg_m_tab,by=c("group","scene"))
...

```{r combo_fig_fish, echo=FALSE}
#### Build Fish Tables
# calculate overall mean percent change in habitat
fish_chg_mean <- fish_pc_tab %>%
  group_by(species) %>%
  group_by(lifestage, add = TRUE) %>%
  group_by(scene, add = TRUE) %>%
  summarise(per_chg = mean(per_chg,na.rm = TRUE)) %>%
  mutate(criteria = ifelse(species %in% native_list,"native","non-native")) %>%
  rename(group = species)
...

#### Display Figures
```{r combo_figs, echo=FALSE, message=FALSE, warning=FALSE}
# plotting fish only - mean
f_plt <- ggplot(data = fish_chg_mean, aes(x = group, y = per_chg,

```

```

    fill = criteria, alpha = lifestage))+
  theme_few()+
  scale_fill_manual(values = c("native"="dodgerblue3", "non-native" = "red2"))+
  geom_bar(stat = "identity", position = "dodge")+
  coord_flip()+
  scale_x_discrete(labels=gsub("_", " ", unique(fish_chg_mean$group)))+
  theme(legend.position = "right",
        text=element_text(size=14, face = "bold", color = "black"),
        axis.text.x = element_text(colour = "black", face="plain"),
        axis.text.y = element_text(colour = "black", face="plain"))+
  facet_wrap(vars(scene), dir = "v")+
  {if(length(lifestages)>1) scale_alpha_discrete(range = c(1,0.35))}+
  {if(length(lifestages)==1) scale_alpha_discrete(guide="none", range = c(1,1))}+
  xlab(element_blank())+
  ylab("% Change in Habitat Area")
# plotting vegetation only
v_plt <- ggplot(data = veg_summary_tab, aes(x = group, y = per_chg, fill =
criteria))+
  theme_few()+
  scale_fill_brewer(palette = "Set1")+
  geom_bar(stat = "identity")+
  coord_flip()+
  theme(legend.position = "right",
        text=element_text(size=14, face = "bold", color = "black"),
        axis.text.x = element_text(colour = "black", face="plain"),
        axis.text.y = element_text(colour = "black", face="plain"))+
  facet_wrap(vars(scene), dir = "v")+
  xlab(element_blank())+
  ylab("% Change in Habitat Area")
plts <- ggarrange(f_plt, v_plt, ncol = 1) # Looks best 8 x 6
plts
...

```

## plot.10day.ts.R

```

# This function will produce 10 day minimum monthly habitat area time series for a
single fish species and lifestage.
# All scenarios are displayed on one plot. Note the units of area are assumed to be
# Last edited by Elaina Passero 6/11/20

plot.10day.ts <-
function(a, species, scene_fish_out, scene_list, NormalizeByL, post_tag){

```

```

# create table of 10-day minimum habitat area that includes all scenarios
ten_day_min <- data.frame()
for(i in 1:length(scene_list)){
  avg_mon_area <- scene_fish_out[[i]][["avg_monthly_area"]][[a]] %>%
    mutate(scene = names(scene_list)[i])
  #mutate(scene_name_full = scene_names_full[i])
  ten_day_min <- bind_rows(ten_day_min,avg_mon_area)
}

# making automatic labeling more readable
ten_day_min_pt <- ten_day_min %>%
  mutate(fill_date = as.Date(paste(year,"-",month,"-15",sep="")))

if(NormalizeByL == "Yes"){
  # plot time series of 10 day min area
  ts_min_plt <-
ggplot(ten_day_min_pt,aes(x=fill_date,y=mon_avg_low_area,color=scene))+
  geom_line()+
  theme_gray()+
  theme(text=element_text(size=12,face = "bold"),
        panel.border = element_rect(fill = NA,size=1),
        legend.justification = c("right","top"),
        axis.text.x = element_text(colour = "black",face="plain",angle =
45,hjust=1),
        axis.text.y = element_text(colour = "black",face="plain"))+
  labs(y=bquote('Minimum 10-day Habitat Area
(Normalized)'),x="Year",title=paste(reach_name,"
",names(species),sep=""),color="")+
  scale_x_date(date_labels = "%Y")
} else{
  # plot time series of 10 day min area
  ts_min_plt <-
ggplot(ten_day_min_pt,aes(x=fill_date,y=mon_avg_low_area,color=scene))+
  geom_line()+
  theme_gray()+
  theme(text=element_text(size=12,face = "bold"),
        panel.border = element_rect(fill = NA,size=1),
        legend.justification = c("right","top"),
        axis.text.x = element_text(colour = "black",face="plain",angle =
45,hjust=1),
        axis.text.y = element_text(colour = "black",face="plain"))+

```



```

      labs(y=bquote('Minimum 10-day Habitat
Area'),x="Year",title=paste(reach_name," ",name(species),sep=""),color="")+
      scale_x_date(date_labels = "%Y")
    }

    ggsave(paste(reach_wd,"dss_outputs/figures/",post_tag,"_",species,"_",a,"_", "10day_h
ab_ts.jpg",sep=""),
           width=7, height=5,units = "in",dpi = 300)
    return(ts_min_plt)
  }

```

## build.10day.tables.R

```

# This script will create summary tables associated with 10-day minimum monthly
# habitat area for each species
# last edited by Elaina Passero on 5/31/20

build.10day.tables <- function(a,species,scene_list,scene_fish_out,post_tag){

  ten_day_min_out <- list()

  ## create table of 10-day minimum habitat area that includes all scenarios
  ten_day_min <- data.frame()
  for(i in 1:length(scene_list)){
    avg_mon_area <- scene_fish_out[[i]][["avg_monthly_area"]][[a]] %>%
      mutate(scene = names(scene_list)[i])
    ten_day_min <- bind_rows(ten_day_min,avg_mon_area)
  }

  ## Create tables of average 10-day minimum monthly habitat summarized by month for
  all years
  sum_all_yrs <- ten_day_min %>%
    dplyr::group_by(scene) %>%
    dplyr::group_by(month,add=TRUE) %>%
    summarize(mon_avg = mean(mon_avg_low_area))
  ten_day_min_out$sum_all_yrs <- sum_all_yrs

  write.csv(ten_day_min_out$sum_all_yrs,file=paste(reach_wd,"dss_outputs/",post_tag,"_",
,species,"_",a,"_10day_min_sum.csv",sep="")) # export table

```

```

# Calculate % change between baseline and scenario habitat: scene-
historic/historic*100
scenes <- scene_list[scene_list != "baseline_q"]
sum_all_yrs_dif <- data.frame()
for(j in 1:length(scenes)){
  scene_name <- names(scenes)[[j]]

  wide_tab <- sum_all_yrs %>%
    filter(scene == "baseline_q" | scene == paste(scene_name)) %>%
    pivot_wider(names_from = scene, values_from = mon_avg)

  per_chg_tab <- wide_tab %>%
    mutate(per_chg = ((wide_tab[[paste(scene_name)]] -
wide_tab[["baseline_q"]])/wide_tab[["baseline_q"]]*100)) %>%
    select(month,per_chg) %>%
    mutate(scene = scene_name)

  sum_all_yrs_dif <- bind_rows(sum_all_yrs_dif,per_chg_tab)
}
ten_day_min_out$sum_all_yrs_dif <- sum_all_yrs_dif

write.csv(ten_day_min_out$sum_all_yrs_dif,file=paste(reach_wd,"dss_outputs/",post_tag
,"_",species,"_",a,"_10day_min_chg.csv",sep=""))

return(ten_day_min_out)
}

```

## make.barplots.chg.R

```

# this function will make barplots of % change from baseline for all species on one
plot
# last edited by Elaina Passero on 5/29/20

make.barplots.chg <-
function(a,species,ten_day_min_outputs,post_tag,scene_names_full){

  # combine all of the sum_all_yrs_dif tables into 1
  agg_sum_df <- ten_day_min_outputs[[a]]$sum_all_yrs_dif %>%
    filter(scene != "baseline_q")

```

```

comp_names <- scene_names_full[-1] # label scenarios getting compared to baseline

# create plots
bp <- ggplot(agg_sum_df,aes(x=month,y=per_chg,fill=scene))+
  geom_bar(position = "dodge",stat="identity")+
  theme_few()+
  scale_fill_brewer(palette = "Set1",name="Scenario",labels = comp_names)+
  theme(legend.position = "bottom",
        text=element_text(size=14,face = "bold",color = "black"),
        panel.border = element_rect(fill = NA,size=1),
        strip.background = element_rect(fill = NA),
        axis.text.x = element_text(colour = "black",face="plain"),
        axis.text.y = element_text(colour = "black",face="plain"))+
  labs(y="% Change in Habitat Area",x="Month",title=species,a)+
  scale_x_continuous(breaks=c(1,2,3,4,5,6,7,8,9,10,11,12),
                    labels =
c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"))

ggsave(paste(reach_wd,"dss_outputs/figures/",post_tag,"_",species,"_",a,"_", "per_chg
.png",sep=""),width=7, height=5,units = "in")
return(bp)
}

```

## id.high.prob.areas.R

```

# This function will identify areas with high probability of occurrence of
vegetation guilds
# Last edited by Elaina Passero 3/2/20

id.high.prob.areas <- function(scene_veg_out,scene_name){
  prob_veg_map <- scene_veg_out[[scene_name]]
  threshold <- 0.7 # threshold to be considered high probability of occurrence

  # select areas of high prob occurrence
  rcl_qual <- data.frame(from = c(-Inf,threshold),
                        to = c(threshold,Inf),
                        becomes = c(0,1)) %>%
    as.matrix(.)

  high_prob_map <- reclassify(prob_veg_map,rcl_qual)
  return(high_prob_map)
}

```

## map.chg.veg.areas.R

```
# This function will map the change in high probability of occurrence areas from
baseline
# Last edited by Elaina Passero on 2/24/20

map.chg.veg.areas <- function(high_prob_areas,scene_name){

  # extract maps
  base_map <- high_prob_areas$baseline_q
  scene_map <- high_prob_areas[[scene_name]]

  diff_map <- scene_map - base_map

  return(diff_map)
}
```

## make.hp.areas.tables.R

```
# This function will make tables of high probability of occurrence areas by
scenario
# Last edited by Elaina Passero on 2/24/20

make.hp.area.tables <-
function(v,high_prob_areas,scene_names,NormalizeByL,reach_length){

  calc_hp_area <- function(one_map){ # calculate high probability area
    cell_size <- xres(one_map)^2
    sum_cells_scene <- cellStats(one_map,stat='sum',na.rm=TRUE)
    if(NormalizeByL == "Yes"){
      area <- data.frame(normalized_area = cell_size*sum_cells_scene/reach_length)
    }else{
      area <- data.frame(total_area = cell_size*sum_cells_scene)
    }
    return(area)
  }

  hp_area_list <- map(high_prob_areas,calc_hp_area)
  names(hp_area_list) <- scene_names
  all_scene_hp_area_df <- bind_rows(hp_area_list,.id="scene")
}
```

```

write.csv(all_scene_hp_area_df, file = paste(reach_wd,
"dss_outputs/",reach_run,"_", "veg", "_", v, "_hp_area.csv", sep = ""), append =
TRUE)

return(all_scene_hp_area_df)

}

```

## calc.chg.veg.area.R

```

# This function will calculate the percent change from baseline in high probability
of occurrence areas
# Last edited by Elaina Passero 2/24/20

```

```

calc.chg.veg.area <- function(hp_area_tab,scene_names,NormalizeByL){

  scenes <- scene_names[scene_names != "baseline_q"]
  per_chg_hp_tab <- data.frame()
  for(j in 1:length(scenes)){
    scene_name <- scenes[j]

    if(NormalizeByL == "Yes"){
      wide_tab <- hp_area_tab %>%
        filter(scene == "baseline_q" | scene == paste(scene_name)) %>%
        pivot_wider(names_from = scene, values_from = normalized_area)

      add_tab <- wide_tab %>%
        mutate(per_chg = ((wide_tab[[paste(scene_name)]] -
wide_tab[["baseline_q"]])/wide_tab[["baseline_q"]]*100)) %>%
        select(per_chg) %>%
        mutate(scene = scene_name)

      per_chg_hp_tab <- bind_rows(per_chg_hp_tab,add_tab)
    } else{
      wide_tab <- hq_area_tab %>%
        filter(scene == "baseline_q" | scene == paste(scene_name)) %>%
        pivot_wider(names_from = scene, values_from = total_area)

      add_tab <- wide_tab %>%
        mutate(per_chg = ((wide_tab[[paste(scene_name)]] -
wide_tab[["baseline_q"]])/wide_tab[["baseline_q"]]*100)) %>%

```

```

      select(per_chg) %>%
      mutate(scene = scene_name)

      per_chg_hp_tab <- bind_rows(per_chg_hp_tab,add_tab)
    }
  } # end of for loop

  return(per_chg_hp_tab)
}

```

## check.veg.movement.R

```

# This script will describe potential vegetation movement
# Last edited by Elaina Passero on 2/24/20

check.veg.movement <- function(wet_map,high_prob_areas,scene_name){
  # find 1st and 3rd quartiles of inundating discharge that high probability area
  occurs in
  hp_iq_map_base <-
  mask(wet_map,high_prob_areas$baseline_q,maskvalue=0,updatevalue=NA)
  hp_iq_map_scene <-
  mask(wet_map,high_prob_areas[[scene_name]],maskvalue=0,updatevalue=NA)

  base_range <- summary(hp_iq_map_base)
  scene_range <- summary(hp_iq_map_scene)

  if(is.na(base_range[1]) == FALSE){ # check that there are high probability of
  occurrence areas

    # describe change in high probability of occurrence areas
    if(scene_range[2] < base_range[2]){
      if(scene_range[4] > base_range[4]){
        movement <- "expand"
      } else{movement <- "encroach"}
    }

    if(scene_range[2] > base_range[2]){
      if(scene_range[4] > base_range[4]){
        movement <- "retreat"
      } else{movement <- "reduce"}
    }
  }
}

```

```

    if(scene_range[2] == base_range[2] && scene_range[4] == base_range[4]){
      movement <- "not change"
    }
  } else{
    if(is.na(scene_range[1]) == FALSE){
      movement <- "encroach"
    }else{
      movement <- NA
    }
  }
}
return(movement)
}

```

## Appendix B: REFDDSSV2 Function Descriptions

Table B.1. Descriptions of functions used in the REFDDSSV2. The primary scripts are pre.processing.R (Module 1), fish.processing.R (Module 2), post.processing.R (Module 3), and scenario.comparison.Rmd (Module 4).

| Function              | Description  |
|-----------------------|--|
| <b>pre.processing</b> | Processes 2D hydraulic modeling results by rasterizing tabulated results or reading in rasters. Rasterizes substrate maps if applicable. Exports a .rdata file including a rasterBrick of 2D hydraulic modeling results for selected |

|                         |   |
|-------------------------|---|
|                         | habitat metrics, a list of modeled discharges, and a rasterized substrate map (if shapefile provided).  |
| <b>get.results</b>      | Pulls in iRIC output results as CSVs and fixes headers containing special characters. Returns a list of CSVs and a list of the modeled discharges.  |
| <b>get.gila</b>         | Reads in SR2DH text file outputs for rasterization. Adds headers and extracts modeled discharges.   |
| <b>iric.process.smr</b> | Rasterizes the results in list of CSVs using attributes of user supplied DEM. Returns a rasterBrick of each habitat metric that contains the results for all modeled discharges.  |
| <b>load.cherry</b>      | Loads Cherry Creek raster files, stored in separate folders by hydraulic habitat variable, into the script. Returns a rasterBrick of each habitat metric that contains the results for all modeled discharges.  |
| <b>fish.processing</b>  | Houses the fish habitat processing options and functionality. Loads inputs from pre.processing script. This script will produce area-lookup tables by species and a list of rasters of habitat sorted by modeled discharge.   |
| <b>find.hsc</b>         | Selects habitat suitability criteria from reach_hsc.csv for the current species.  |
| <b>find.sub</b>         | find.sub: Selects substrate criteria from reach_substrate.csv for the current species.  |
| <b>bricks.rc</b>        | Reclassifies rasterBricks of 2D hydraulic modeling results into maps of hydraulically suitable habitats by species and lifestage based on habitat suitability criteria (HSC csv). Returns a Brick of suitable cells by discharge.   |
| <b>by.substrate</b>     | Takes a rasterized substrate type map of the same extent, resolution, and projection of DEM and reclassifies it into suitable or unsuitable substrate based on user supplied substrate criteria table. Uses the reclassified substrate map as a mask over the Brick of hydraulically suitable cells to extract cells that have suitable habitat. Returns a rasterBrick of suitable habitat. |
| <b>remove.islands</b>   | Removes islands (small, isolated habitat areas) from suitable habitat area. The number of cells that constitutes an   |



|                               |  |
|-------------------------------|--|
|                               | island can be specified in the fish.processing script. Follow's rook's rules (4 sides) to determine clumps of cells.   |
| <b>total.area</b>             | Calculates total available and, optionally, reach-normalized habitat area by lifestage for the area-discharge reference tables. This function will also export the habitat area-discharge lookup tables to the dss_outputs folder if that option has been specified. Area is calculated by multiplying the number of suitable cells by the cell size (taken from raster resolution).   |
| <b>order.by.q</b>             | Puts rasters of total available habitat in order of ascending modeled Q for easy extraction. This function will export the rasters to the dss_outputs folder if that option has been specified. This function is also called on in make.inun.q.map.R (post.processing) to order rasters in out_val_rast by modeled discharge to make inundating discharge maps.  |
| <b>plot.a.q.curve</b>         | This function will plot habitat area-discharge curves for all modeled discharges. Plots are saved as .pngs in the reach working directory.   |
| <b>plot.a.q.curve.in.chan</b> | This function will plot habitat area-discharge curves for all in-channel modeled discharges. The threshold for in-channel discharge must be set inside the function. The filter() function takes this input. Plots are saved as .pngs in the reach working directory.  |
| <b>post.processing</b>        | This script delivers outputs relating both fish and riparian vegetation habitat to alternative flow scenarios. Multiple flow scenarios can be provided to the script via CSVs. All species, guilds, lifestages, etc. can be run at the same time for a single reach.   |
| <b>interp.table</b>           | Adds total available and normalized area (if applicable) to the hydrograph table. It will linearly interpolate between discharges to fill in areas for any discharges not run in iRIC. This function is not capable of extrapolation. Currently, a discharge of 0 is added to the tables with 0 total habitat area (and normalized area if applicable) to address flows below the minimum modeled discharge. In the future these discharges will be modeled. |

|                         |  |
|-------------------------|--|
| <b>interp.plot</b>      | This function generates plots of total available and normalized habitat area (if applicable) with the hydrograph. Plots are saved as .pngs in the working environment.   |
| <b>x.day.stats</b>      | Calculates moving X-day minimum, maximum, and mean discharges and total area for a hydrograph.   |
| <b>avg.month.area</b>   | Calculates the minimum average monthly area using a central running 10-day mean. The 10-day mean is calculated for every day excluding the first four and last five days in the record. The lowest value in a month is selected as the minimum average monthly area. This value can include days from the previous, current, or next month.              |
| <b>interp.plot.abbr</b> | This function generates plots of total available and normalized habitat area (if applicable) with the hydrograph for a subset of time. Plots are saved as .pngs in the working environment.  |
| <b>q.ep.weibull</b>     | This function will calculate exceedence probability (EP) for flows in flow scenario and any modeled discharges above the maximum flow in the scenario. If flows above the maximum scenario flow are input, EP will be calculated from Weibull plotting position assuming 1 event per discharge. These flows need an EP value to appear on the map of EP. |
| <b>make.inun.q.map</b>  | Produces map of inundating discharge for each cell. This map is produced by using the cover (raster package) function to label cells by the modeled discharge where they become wet (depth > 0). GeoTiffs are exported to dss_outputs.   |
| <b>make.ep.maps</b>     | Reclassifies inundating discharge map with EP values from a flow scenario. EP values vary with flow scenario, so EP for the same flow may be mapped to different areas in different scenarios. GeoTiffs are exported to dss_outputs.   |
| <b>find.veg.logit</b>   | This function pulls riparian vegetation logistic regression equations for a single group from the full table of logistic regression equations.   |
| <b>find.prob.occur</b>  | This function will calculate probability of occurrence of vegetation groups from EP. Values for the coefficients are assigned by searching for phrases starting with "B". Coefficients are assumed to follow the series "B0", "B1", .... Logistic regression equations are parsed at every "*x" in the   |

|                            |   |
|----------------------------|---|
|                            | equation. The X's are replaced with EP values. This function only works with one independent variable X.  |
| <b>make.veg.maps</b>       | This script will make maps of probability of occurrence of vegetation groups by reclassifying EP maps. GeoTiffs are exported to the working environment.  |
| <b>scenario.comparison</b> | This markdown script will generates tables and figures to support scenario comparison and tradeoff evaluation.  |
| <b>plot.10day.ts.R</b>     | Will produce minimum 10 day average monthly area time series for a single species.  |
| <b>build.10day.tables</b>  | This script will create summary tables associated with 10-day minimum monthly habitat area for each species. Outputs include long tables of area that includes all scenarios, tables of area summarized by month for all years, and tables of percent change between baseline and the scenario. |
| <b>make.barplots.chg</b>   | This function will make barplots of percent change from baseline for all scenarios for a single species and life stage. Plots are exported to the working environment as jpgs.  |
| <b>id.high.prob.areas</b>  | This function will identify areas with high probability of occurrence of vegetation guilds. The threshold for "high probability" is currently set at 0.7.   |
| <b>map.chg.veg.areas</b>   | This function will map the change in high probability of occurrence areas from baseline.  |
| <b>make.hp.area.tables</b> | This function will make tables of the area of high probability of occurrence in each scenario. Tables are specific to each guild/species.   |
| <b>calc.chg.veg.area</b>   | This function will calculate the percent change from baseline in high probability of occurrence areas. A table is made for every species giving the percent change by scenario.   |
| <b>check.veg.movement</b>  | This function will describe potential vegetation movement based on the 1 <sup>st</sup> and 3 <sup>rd</sup> quartiles of the distribution of inundating discharges that high probability of occurrence areas are found in.   |

## Appendix C: REFDSSV2 Quick Start Tutorial

This document is designed to introduce users to the REFDSSV2 by walking them through the process to run a single reach.

### Instructions for Adding Reach Folders for DSS

#### Adding New Reach

- In the reaches folder create a new folder with the name of the reach followed by an underscore and a unique tag. This tag is used to distinguish sets of 2D hydraulic modeling results from each other.
  - For example USBeasley1\_2.12.20
- In this folder add five folders: model\_outputs, habitat\_info, dss\_outputs, temp, and flow\_scenarios.
- Add the shapefile of substrate types to the reach folder if you have that file.
- The temp folder should be left empty. It holds temporary files used during rasterization and .RData file recall.
- Inside model\_outputs add the 2D hydraulic modeling outputs and the TIFF of the DEM of the reach. The 2D hydraulic modeling outputs may be added as CSVs directly into the model\_outputs folder or as externally rasterized results. CSVs must be named with the modeled discharge they hold results for. It is important that discharges are indicated numerically as the DSS cannot interpret a combination of numbers and letters (i.e. using “k” to denote 1,000). Externally rasterized results can be used (LoadExternal set to “Yes”), but it is the users’ responsibility to figure out how to get them into R.
- Add CSVs of fish habitat suitability criteria, vegetation preference equations, and fish substrate requirements (if applicable) to habitat\_info. These files should be labeled as follows:
  - Fish habitat suitability criteria: ReachName\_hsc
  - Fish substrate requirements: ReachName\_substrate
  - Vegetation logistic regression preference equations: ReachName\_veg\_pref
  - See Sample Reach for formatting guidelines
- In the flow\_scenarios folder, add time series of flow scenarios using the naming structure ReachName\_scenario\_name.CSV. In order to run Module 3, a historic flow record must be supplied under the name ReachName\_baseline\_q.CSV. Basic flow scenarios can be created in make.flow.scenarios.R.
- Inside dss\_outputs add two folders: figures and internal. These folders should be left empty as they will be populated as the DSS runs.

#### Adding New Model Run of Existing Reach

- In the reaches folder create a new folder with the name of the reach followed by an underscore and a unique tag. This tag is used to distinguish sets of 2D hydraulic modeling results from each other.
- In this folder add three new folders: model\_outputs, dss\_outputs, and temp.
- The temp folder should be left empty. It holds temporary files used during rasterization and .RData file recall.
- Inside model\_outputs, add the new set of 2D hydraulic modeling outputs and the TIFF of the DEM of the reach.

- Inside dss\_outputs add two folders: figures and internal. These folders should be left empty as they will be populated as the DSS runs.

## Initial Setup

### Pulling DSS from Github

- Begin by downloading the entire verde-refdss repository from Github. This repository contains all of the scripts and functions needed to run the DSS. The repository can be found at <https://github.com/passeroe/verde-refdss>.
  - Unzip the folder (verde-refdss-master.zip) if it was downloaded from the repository.
- Open R Studio (using 3.6.1 as of 1/23/20). Under File click Open Project. Go into the repository that was unzipped and open verde-refdss.Rproj. Opening this project will take you into the DSS.

### Preparing to Run Existing Reach

- Choose a reach to run from those listed in the reaches folder in the repository or make a copy of an existing reach folder if you are uploading new/updated results and append a unique tag (with no spaces or special characters) to the folder to distinguish the different outputs (USBeasley1\_20200127). Dates in YYYYMMDD or a similar format work well as tags. An underscore should be used to distinguish the reach name from the tag.
- This tutorial will work with the upstream site at Beasley Flats on the Verde River called USBeasley1.
- Open the USBeasley1 folder. Create two folders: dss\_outputs model\_outputs.
- In the model\_outputs folder, add the CSVs containing the 2D hydraulic modeling results and the TIFF of the DEM of the reach.
- In dss\_outputs add folders called figures and internal. Do not add anything to dss\_outputs or figures. These will be populated by fish.processing.R and post.processing.R.
- All of the required habitat suitability information and flow scenarios are downloaded with the repository in a folder called habitat\_info.

## Pre-Processing

### Setting Inputs and Options in pre.processing.R

- Open the pre.processing.R script. Under Primary Inputs change wd, the working directory, to the folder path containing the DSS repository. Keep this string in quotes. Note that a single forward slash should be used between folders. There should be a forward slash at the end of the working directory.
- Set the variables to the values listed in the table. Values should be pasted exactly as written (case-sensitive) after the "<-" for each variable.
- Do not change any variables after "#### End of User Inputs ###".
- Save the changes to the script.
- Select all of the text from line 1 to "#### End of User Inputs ###". While the text is selected hit the Run button in the top right of the window.
- Running this section will install and load all required packages and define the inputs.

Table C.1. Inputs for pre.processing.R for tutorial.

| Variable Name       | Description  | Value                                  |
|---------------------|--|--|
| <b>hab_mets</b>     | habitat metrics                                      | list("Depth","Velocity")               |
| <b>reach_name</b>   | reach name   | "USBeasley1"                           |
| <b>model_run</b>    | Identify the modeling results                        | "2.12.20"                              |
| <b>dis_unit</b>     | unit of modeled discharges                           | "cms"                                  |
| <b>LoadExternal</b> | load external variables                              | "No"                                   |
| <b>skip_num</b>     | number of rows to skip in tabulated modeling results | 0                                      |
| <b>x_loc</b>        | field name of X coordinate in model outputs          | "X"                                    |
| <b>y_loc</b>        | field name of Y coordinate in model outputs          | "Y"                                    |
| <b>dem</b>          | TIFF of digital elevation model                      | "BeasleyUS_0.25m_20191108.tif"         |
| <b>setRes</b>       | manually set resolution of rasterized results        | "Yes"                                  |
| <b>res</b>          | resolution of rasters                                | c(0.5,0.5)                             |
| <b>CheckSub</b>     | check substrate for habitat suitability              | "Yes"                                  |
| <b>sub_name</b>     | name of substrate map shapefile                      | "BeasleyUS_SedThiessenPoly1 Dissolved" |
| <b>sub_field</b>    | field containing substrate type                      | "ParticalSi"                           |

#### Running pre.processing.R

- Select lines from "#### Begin Processing ####" to the end of the script.
- While the text is selected, hit Run in the upper right corner of the window.
- Rasterizing the CSV outputs is time intensive. If this process is taking too long, hit Stop and rerun with a larger raster resolution.
- Outputs will be stored in a file called USBeasley1\_pre\_outputs.rdata.

## Instream Habitat Processing

Setting Inputs and Options in fish.processing.R

- Open fish.processing.R script. Under Primary Inputs change wd, the working directory, to the folder path containing the DSS repository. Keep this string in quotes. Note that a single forward slash should be used between folders. There should be a forward slash at the end of the working directory.
- Set the variables to the values listed in the table. Values should be pasted exactly as written (case-sensitive) after the "<-" for each variable.
- Do not change any variables after "### End of User Inputs ###".
- Save the changes to the script.
- Select all of the text from line 1 to "### End of User Inputs ###". While the text is selected hit the Run button in the top right of the window.
- Running this section will install and load all required packages and define the inputs.

Table C.2. Inputs for fish.processing.R for tutorial.

| Variable Name         | Description                             | Values                             |
|-----------------------|---|------------------------------------|
| <b>fish_tag</b>       | Unique code identifying run             | "tutor"                            |
| <b>hab_mets</b>       | habitat metrics                         | list("Depth","Velocity")           |
| <b>species_list</b>   | species list                            | c("desertsucker","smallmouthbass") |
| <b>lifestages</b>     | life stages                             | list("adult")                      |
| <b>reach_name</b>     | reach name                              | "USBeasley1"                       |
| <b>model_run</b>      | Identify the modeling results           | "2.12.20"                          |
| <b>dis_unit</b>       | unit of modeled discharges              | "cms"                              |
| <b>Check0Flow</b>     | calculate habitat for 0 discharge       | "No"                               |
| <b>depth0Flow</b>     | raster of depth at 0 discharge          |                                    |
| <b>CheckSub</b>       | check substrate for habitat suitability | "Yes"                              |
| <b>RemovesIslands</b> | remove habitat islands                  | "Yes"                              |
| <b>island_size</b>    | island size threshold                   | 2                                  |
| <b>NormalizeByL</b>   | normalize area by reach length          | "Yes"                              |
| <b>reach_length</b>   | reach length                            | 0.5                                |
| <b>length_unit</b>    | reach length unit                       | "km"                               |
| <b>ExportRast</b>     | export habitat rasters                  | "Yes"                              |

### Running fish.processing.R

- Select lines from “#### Begin Processing ####” to the end of the script.
- While the text is selected, hit Run in the upper right corner of the window.
- This section of the script may take a while (+2 hrs) to run. The raster layers are large and there are two (one for each habitat metric) for every species for every discharge.
- Once the script has finished running, plots of the habitat area-discharge curves can be viewed by typing `area_look_plots` into the console and hitting enter. The curves should match this plot.
- Check the `dss_outputs` folder to ensure results have been generated. There should be Tiffs of habitat area for each species for every discharge and CSVs containing the habitat area-discharge lookup tables. Area-discharge curves can be found in the figures folder.
- Outputs will be stored in a file called `USBeasley1_fish_outputs.rdata`.

### Time Series Analysis

#### Setting Inputs and Options in post.processing.R

- Open `post.processing.R` script. Under Primary Inputs change `wd`, the working directory, to the folder path containing the DSS repository. Keep this string in quotes. Note that a single forward slash should be used between folders. There should be a forward slash at the end of the working directory.
- Set the variables to the values listed in the table. Values should be pasted exactly as written (case-sensitive) after the “<-” for each variable.
- Do not change any variables after “#### End of User Inputs ####”.
- Save the changes to the script.
- Select all of the text from line 1 to “#### End of User Inputs ####”. While the text is selected hit the Run button in the top right of the window.
- Running this section will install and load all required packages and define the inputs.

#### Running post.processing.R

- Select lines from “#### Begin Processing ####” to the end of the script.
- While the text is selected, hit Run in the upper right corner of the window.
- Navigate to the ‘figures’ folder where you can find habitat area through the hydrograph and barplots of percent change.
- The `dss_outputs` folder should contain tables of the minimum 10-day average areas, exceedance probability maps, and probability of occurrence of vegetation maps.
- These maps can be viewed in R by using the following format:
  - `plot(scene_veg_out$'scenario name'$'map type'$'vegetation group')`
  - Start by typing `plot(scene_veg_out$` into the console. Replace scenario name, map type, and vegetation group. Suggestions should pop up after typing a ‘\$’.



Table C.3. Inputs for post.processing.R for tutorial.

| <b>Variable Name</b> | <b>Description</b>                                  | <b>Values</b>                      |
|----------------------|---|------------------------------------|
| <b>post_tag</b>      | Unique code identifying post.processing run         | "tutor"                            |
| <b>fish_tag</b>      | Unique code identifying fish.processing run         | "tutor"                            |
| <b>hab_mets</b>      | habitat metrics                                     | list("Depth","Velocity")           |
| <b>species_list</b>  | species list  | c("desertsucker","smallmouthbass") |
| <b>lifestages</b>    | life stages   | list("adult")                      |
| <b>veg_list</b>      | vegetation list                                     | c("drgt_tol_shrubs","tall_trees")  |
| <b>reach_name</b>    | reach name  | "USBeasley1"                       |
| <b>model_run</b>     | Identify the modeling results                       | "2.12.20"                          |
| <b>dis_unit</b>      | unit of modeled discharges                          | "cms"                              |
| <b>date_form</b>     | Format of dates in scenario                         | "%/%d/%Y"                          |
| <b>CheckSub</b>      | was substrate part of habitat suitability criteria? | "Yes"                              |
| <b>NormalizeByL</b>  | normalize area by reach length                      | "Yes"                              |
| <b>reach_length</b>  | reach length  | 0.5                                |
| <b>length_unit</b>   | reach length unit                                   | "km"                               |
| <b>CalcXDayStats</b> | calculate X-day statistics                          | "No"                               |
| <b>x_days</b>        | number of days in trailing window                   |                                    |
| <b>DateRange</b>     | limit analysis to a date range                      | "No"                               |
| <b>start_date</b>    | start date for subset                               |                                    |
| <b>end_date</b>      | end date for subset                                 |                                    |
| <b>LoadScenario</b>  | load external flow scenarios                        | "Yes"                              |
| <b>scene_names</b>   | names of flow scenarios to load                     | "scenario_A"                       |

## Scenario Comparison

Setting Inputs and Options in scenario.comparison.Rmd

- Open scenario.comparison.Rmd script. In the “## User Inputs” block change wd to the folder path containing the DSS repository. Keep this string in quotes. Note that a single forward slash should be used between folders. There should be a forward slash at the end of the working directory.
- Set the variables to the values listed in the table. Values should be pasted exactly as written (case-sensitive) after the “<-” for each variable.
- Do not change any variables outside the User Inputs block
- Save the changes to the script.
- Click Knit then Knit to HTML or Run then Run All to run the markdown script.

Table C.4. Inputs for scenario.comparison.Rmd for tutorial.

| <b>Variable Name</b> | <b>Description</b>                                  | <b>Values</b>                      |
|----------------------|---|------------------------------------|
| <b>post_tag</b>      | Unique code identifying post.processing run         | "tutor"                            |
| <b>fish_tag</b>      | Unique code identifying fish.processing run         | "tutor"                            |
| <b>hab_mets</b>      | habitat metrics                                     | list("Depth","Velocity")           |
| <b>species_list</b>  | species list  | c("desertsucker","smallmouthbass") |
| <b>lifestages</b>    | life stages   | list("adult")                      |
| <b>veg_list</b>      | vegetation list                                     | c("drgt_tol_shrubs","tall_trees")  |
| <b>reach_name</b>    | reach name  | "USBeasley1"                       |
| <b>model_run</b>     | Identify the modeling results                       | "2.12.20"                          |
| <b>dis_unit</b>      | unit of modeled discharges                          | "cms"                              |
| <b>date_form</b>     | Format of dates in scenario                         | "%/%d/%Y"                          |
| <b>CheckSub</b>      | was substrate part of habitat suitability criteria? | "Yes"                              |
| <b>NormalizeByL</b>  | normalize area by reach length                      | "Yes"                              |
| <b>reach_length</b>  | reach length  | 0.5                                |
| <b>length_unit</b>   | reach length unit                                   | "km"                               |
| <b>CalcXDayStats</b> | calculate X-day statistics                          | "No"                               |
| <b>x_days</b>        | number of days in trailing window                   |                                    |
| <b>DateRange</b>     | limit analysis to a date range                      | "No"                               |
| <b>start_date</b>    | start date for subset                               |                                    |
| <b>end_date</b>      | end date for subset                                 |                                    |
| <b>LoadScenario</b>  | load external flow scenarios                        | "Yes"                              |
| <b>scene_names</b>   | names of flow scenarios to load                     | "scenario_A"                       |