Energy Research and Development Division
FINAL PROJECT REPORT

MODELING TOOL TO ASSESS AND MITIGATE THE EFFECTS OF SMALL HYDROPOWER ON STREAM FISHES IN CHANGING CALIFORNIA CLIMATE

Appendix A: Weaphish User's Manual: A Guide To Installing And Running A Habitat-Based Fish Population Model, V. 0.1

Prepared for: California Energy Commission
Prepared by: University of California, Davis

JUNE 2016
CEC-500-2016-043-APA

WEAPhish User's Manual

A guide to installing and running a habitat-based, fish population model, v. 0.1 beta

D. Patrick Kilduff[1], Lisa C. Thompson[1], Samuel S. Sandoval[2], Sooyeon Yi[2]

University of California Davis
1. Department of Wildlife, Fish, and Conservation Biology and 2. Department of Land, Air, and Water Resources

March 31, 2015

Table of Contents

WEAPhish User's Manual 3

Acknowledgements 5

Introduction 7

- Background 7

Frequently Asked Questions 8

Definitions 8

Instructions 9

- Install Python 9
- Preparing Input Data Files 11
  - species_info.txt 12
  - control_options.txt 13
  - control_process.txt 13
  - spawn.txt 15
  - supplement.txt 16
  - relations.txt 17
  - wua.txt 22
  - wua_stage_scalars.txt 23
  - stream.txt 24
  - temps.txt 26
  - flows.txt 26
- WEAPhish code files and simulation 27
- State-variable Arrays 29
- Function Definitions 30
- Excel "middle-ware" for running WEAP and WEAPhish 31
- Output Data Files 32
- Data visualization 37

WEAPhish User's Manual

A guide to installing and running a habitat-based, fish population model, v. 0.1 beta.

This manual guides prospective users through the process of installing the software require to run WEAPhish. We cover preparing input data files, running the model, key properties of model functions, internal data structures, & output data, as well as options for viewing stream habitat and simulated fish population data. WEAPhish uses spatially and temporally explicit temperature and flow data for the selected study stream to model single-species demographic processes of stream fishes (currently implemented for an anadromous salmon). The model's name reflects efforts to integrate it with the water resource planning software WEAP, Water Evaluation and Planning systmem (http://www.weap21.org), developed by the Stockholm Environment Institute (http://www.sei-international.org/us).

WEAPhish is a Python implementation of the USGS Salmod model (Bartholow et al. 1993, 2002). We choose this path because, in part, we desired to continue past investigations based on Salmod, which is no longer supported by the USGS. This allows us to compare WEAPhish results from existing Salmod results from common model configurations, but allows for future development to expand existing model functionality. We wrote WEAPhish in Python because the language has a relatively low-learning curve (www.python.org) and has excellent support for scientific computing and simulation (see http://www.scipy.org/).

WEAPhish runs at the command line as a Python script or via an Excel VBA Spreadsheet (Master_WEAPhish) that serves as "middle-ware" between WEAP and WEAPhish. WEAPhish will run on Windows, OS X, or Linux, but requires that Python is installed on your computer along with additional Python modules (i.e., libraries of functions that provide additional computational functionality and data structures for conducting scientific simulation and analyses – these libraries are detailed in Instructions: Install Python). WEAPhish was written in Python 2.7.9. ***Do not use Python 3***. The easiest way to install Python and the additional modules required for WEAPhish

is to install a scientific distribution of Python (see http://www.scipy.org/install.html). We used the open source Continuum Analytics Anaconda Python distribution (https://store.continuum.io/cshop/anaconda/) .

One can also install Python (www.python.org) and the the necessary modules individually (please see https://docs.python.org/2.7/installing/index.html for details on module installation). It is not necessary to be able to program in Python to run the model. The WEAPhish model code is available from contacting the WEAPhish programmer (Patrick Kilduff: dpkilduff@ucdavis.edu.

Acknowledgements

We thank Ken Krich, Research Coordinator and Assistant Director, California Institute for Energy and Environment, Eric Lee, CIEE Research Administrator, Lillian Mirviss, California Energy Commission Contract Manager, and retired CEC Contract Manager Joseph O'Hagan, for their support and assistance at all stages of this project.

This project is indebted to the USGS and the developers of the SALMOD program. We hope that we can continue and advance the line of research and simulation modeling

WEAPhish is a step towards continuing and expanding cross-disciplinary research efforts that began in collaboration with our colleagues at Stockholm Environment Institute-US (SEI-US) and UC Davis (UCD): David Purkey (SEI-US), Jack Sieber (SEI-US), Marisa Escobar (SEI-US), Josue Medellin-Azuara (SEI-US), and Christopher Mosser (UCD).

We appreciate the time and input from several stakeholders and collaborators who gave their time and provided input on modeling needs and insights on the ecology, hydrology, and management of stream systems in central California: Ed Cheslak (Pacific Gas & Electric), Marisa Escobar (SEI-US), Anna Ewing (California Department of Fish and Wildlife), David Hanson (Sacramento Municipal Utility District), Allen Hawthorn (Friends of Butte Creek), Sean Hoobler (California Department of Fish and Wildlife), Beth Lawson (California Department of Fish and Wildlife), Michael Maher (CA State Water Resources Control Board), Tracy McReynolds (California Department of Fish and Wildlife), Susan Monheit (CA State Water Resources Control Board), Darold Parry (Sacramento Municipal Utility District), David Purkey (SEI-US), Catalina Reyes (Pacific Gas & Electric), Chris Shutes (California Sportfishing Protection Alliance), Michael Smith (Friends of Butte Creek), Dan Teeter (US Forest Service), Gretchen Umlauf (NOAA Fisheries), Amber Villalobos (CA State Water Resources Control Board), Jeff Wetzel (CA State Water Resources Control Board), and Chuck Young (SEI-US).

Finally, we thank the UC Davis, CIEE, and CEC administrative staff who have supported and facilitated our efforts.

Introduction

Background

Competing uses for a limited supply of surface water in California combined with natural variability of environmental factors influencing water supply and biological productivity of stream dwelling fishes pose challenges for managers and policy makers. Such challenges require tools that help inform water management decisions by quantifying trade-offs inherent in complex social-economic systems. One particular trade-off in water management is how climate potential climate change influences the extinction risk of protected salmon populations on streams on streams managed for other purposes such as hydropower. While streams with (small) hydropower systems impact salmon, modifying hydropower operational rules have the potential to lengthen the persistence of salmon threatened by stream warming (Thompson et al. 2012). Evaluating the trade-offs between renewable energy generation and persistence of ecologically, socially, and economically important fish species, allows resource managers to probe for policies that achieve desirable results, while being robust to uncertainty of the underlying ecosystem processes.

Below, we describe our "reincarnated" version of the USGS's Salmod model – a spatially explicit habitat based population model designed to examine the influence of stream temperature and flow on demographic processes of stream fishes (currently designed for anadromous fishes only). While Salmod is still freely available from the USGS, it is no longer actively supported. We re-implemented Salmod in order to build on existing research that improved our understanding of the intertwined relationships among fish productivity, climate change, and hydropower operation. First, this approach allows us to replicate previous analyses. Second, we now have the ability to expand model functionality to explore different model configurations (e.g., model the entire

salmon life cycle – Including simple accounting for ocean survival or run simulations at finer (i.e., daily) time steps). Third, we developed a formal link between a fish-habitat model and the water resource planning tool WEAP, which was achieved by developing middle-ware that allows users to control both WEAP and WEAPhish from user friendly confines of a macro-enabled Excel workbook. Linking these two models facilitates integrated analyses of trade-offs involved in water resource allocation between protected fish species and hydropower operations.

Frequently Asked Questions

1. Why revise the Salmod model? Because we have used Salmod in earlier investigations, we believe it is practical to start our own line of model development based on an existing peer-reviewed modeling framework. Salmod is no longer supported and only available for Windows (32-bit) machines. Re-writing Salmod as WEAPhish will allow us and others to continue model development (i.e., adding features, improving performance and ease of use), testing, and application.

2. Where can I get WEAPhish? From the author: Patrick Kilduff: dpkilduff@ucdavis.edu. Once the model is peer-reviewed, users will be able to download the model from a source code repository (either BitBucket or GitHub).

3. Some of the input files appear to count from 0 in some places, why is that? Indices for Python data structures (i.e., lists, vectors, arrays, Pandas DataFrames) begin at 0. In case the user “pokes” around the model code, please keep this in mind.

Definitions

This section contains definitions of selected terms that have specific meaning as used in the model and user’s manual.

Computational unit: An individual stream segment defined by its position, length, and habitat type of the stream study system. Often “comp_unit” or “cu” in model code.

Group (cohort): Refers to a collection of fish of the same stage and class within a computational unit. Only one group of each adult stage (adult males/females, spawning males/females) is allowed, but there may be multiple groups of egg and juvenile stages within a given computational unit. A new group of eggs, for example, is created each week in each computational unit during the spawning season. New groups of juveniles are created when members of a group must move to a new computational unit. If the maximum number of groups is exceeded in a computational unit, the model will find two groups of the same stage and size class to merge. This creates an “open” group slot in the computational unit. The number of groups is limited to constrain the computational burden.

Instructions

The steps necessary to run WEAPhish are described below.

Install Python

We used the Continuum Analytics, Inc. Anaconda distribution of Python (version 2.1.0) and describe the steps to install this distribution of Python (https://store.continuum.io/cshop/anaconda/). Like other scientific distributions of Python, Anaconda contains the primary Python modules for scientific computing (e.g., Numpy, Scypy, Pandas, Matplotlib, etc.) plus many (100s) of other modules. Other distributions of Python will work (http://www.scipy.org/install.html), but the user will need to ensure that the following packages are installed or install them separately:

- Numpy (http://www.numpy.org/)
- Pandas (http://pandas.pydata.org/)
- Scipy (http://www.scipy.org)
- Numba (http://numba.pydata.org/index.html)
- Dateutil (http://labix.org/python-dateutil)

Other modules used in WEAPhish that should be available from the base Python installation are: shlex, datetime, re, and os.

The best way to install the Anaconda distribution (or the other scientific distributions) of Python is to visit their website (https://store.continuum.io/cshop/anaconda/) and follow the installation instructions for your operating system. Once Python and the required packages are installed, open text editor (e.g., TextEdit on OS X or Notepad on Windows) and write the following lines (I don't suggest copy and pasting these lines):

```
#! usr/bin/env python
# -*- coding: utf-8 -*-

import pandas
import numpy
import scipy
import numba
import shlex
import datetime
import re
import os
import datetime
import dateutil

print "Success! Imported all modules used by WEAPhish."
```

And save this file as "python_mod_test.py" (on your Desktop). When this Python script is run, it will load all the packages used in WEAPhish if they are installed. If you're on OS X or a Linux machine open a Terminal window. On OS X go to Applications/Utilities and double-click on "Terminal".

At the Terminal prompt type (don't type the ">", that's the default prompt):

```
> cd ~/Desktop
```

and press "return" to change the working directory to Desktop. Then at the Terminal prompt, type:

```
> python python_mod_test.py
```

and press "return". If all the packages load correctly, "Success! Imported all modules used by WEAPhish." will display in the Terminal window above a new Terminal prompt.

On Windows click on the Window start button, then type "command prompt" in the search bar. Select "Command Prompt" and press return. At the Command Prompt type:

```
> chdir Desktop
```

and press “Enter” to change the working directory to Desktop. Then at the Command Prompt, type:

```
> python python_mod_test.py
```

and press “Enter”. If all the packages load correctly, “Success! Imported all modules used by WEAPhish.” will display in the Command Prompt window above a new prompt.

If the packages do not load correctly, then Python will return a descriptive error that identifies the (first) module that failed to load successfully – for example: “ImportError: No module named dateutil”. If that occurs the named module likely did not load correctly or did not load. If you get an error indicating that a module is not installed, please check https://docs.python.org/2.7/installing/index.html and the website for that package or the Python distribution you’re using.

Preparing Input Data Files

With Python installed the next step to running the model is to prepare the input data files. Ideally, all input data for each species and study stream would be drawn from research on the study system and species. However, data will often have to be borrowed from similar species in the same stream or the same or similar species from similar streams. In other cases, expert opinion may be required. (Make note of the source of the data and conduct sensitivity analyses, especially for the more uncertain the input data). Because WEAPhish is based on the Salmod modeling framework, the names, format, and purpose of the input files are very similar to those used for Salmod. That is, researchers who have used Salmod in the past should find it straightforward to convert their existing Salmod input files for use in WEAPhish. WEAPhish has 12 input files. The format and purpose of each is described in detail below.

Input data are written in tab-delimited .txt files. The files can be created/edited using either a text editor or in Excel (save as .txt with tab-delimiters). Most users will likely find it easiest to use Excel to edit the input data, because it is easier to examine and edit the columns of input data in Excel. (Users opting to run the WEAPhish from the Excel middle-ware can enter the input data using that workbook’s macro-assisted interface.)

WEAPhish requires detailed information on the stream habitat (flows, temperature) at fine spatial (100s of meters) and time (weekly) scales. In addition, data describing the growth, mortality, and movement of each stage/size class defined for the study species are required, as is information on the weighted usable area and habitat type for each computational unit of the stream. All input data are described below in detail. Please contact Patrick Kilduff at dpkilduff@ucdavis.edu if you have any questions.

species_info.txt

This file describes the stage and size class information for the study species. The first column “species” contains the species name. The second column “realm” contains “FW” if the stage/size class occurs in the study stream (i.e., freshwater) or “O” if it occurs outside of the study stream (i.e., ocean). Currently, the model only requires (and works) for FW stages. Implementing a model that accounts for survival during the ocean phase of salmonids is planned in future model development. The third and fourth columns contain the “stage” name, size “classes” index, respectively. Note: In Python, “class” is a reserved word not to be used to name other objects, so variants of “class” like “cls” or “classes” are used to prevent errors. The stage names and stage indices are fixed, but for egg, fry, fingerling and juvenile stages up to five user-specified size classes may be specified. The fifth column contains the integer index corresponding to the “stage” name in column 3. Note that adult and spawner stages have a single size class and that they do not have any size related information. Size information for adults is provided in the supplement.txt file. The sixth and seventh columns contain the size range (“minimum” and “maximum”) for each stage and size class. For stages “fry”, “fingerling”, and “juvenile” the size data represent the minimum and maximum length (mm). For the “egg” stage, minimum and maximum values correspond to the percent maturation of a cohort. The columns “mean” and “sd” are not currently used.

FIGURE species_info.txt: Example structure of the input file "species_info.txt".

	C	D	E	F	G
1	stage	classes	stg_idx	minimum	maximum
2	adult_female	1	0	NA	NA
3	adult_male	1	1	NA	NA
4	spawning_fem	1	2	NA	NA
5	spawning_mal	1	3	NA	NA
6	eggs	1	4	0	0.6
7	eggs	2	4	0.6	1
8	fry	1	5	25	40
9	fry	2	5	40	60
10	fingerling	1	6	60	80
11	fingerling	2	6	80	100
12	juvenile	1	7	100	125
13	juvenile	2	7	125	150
14	juvenile	3	7	150	250

control_options.txt

This file contains basic information that the fish model uses to define the length of an annual model run. The first column contains variable names beginning on row 2 and the second column contains the values assigned to the variable. This file contains the number of annual time steps, the simulation start date ("begin_date"), specifies if habitat "capacity" is measured in terms of numbers of fish or biomass, the "population" and "model" specify configurations. Currently, the only valid values for these control variables are "population" equals "anadromous" and "model" equals "escapement".

FIGURE control_options.txt.

	A	B
1	variable	value
2	timesteps	52
3	begin_date	5/25/10
4	capacity	numbers
5	population	anadromous
6	model	escapement

control_process.txt

WEAPhish is a population dynamic model. That means that WEAPhish simulates demographic processes (e.g., growth, maturity/spawning, movement, and mortality) and

tracks the fate of the population of fish studied over space and time. As fish population models go, WEAPhish has both fine temporal (weekly,) and spatial (several hundred meters) scales.

The file control_process.txt specifies order of the demographic and accounting processes that operate each time step and specifies specific options for each of those processes. Thus, this file contains the weekly instructions for determining the order that a group fish advances to the next life-stage (based on their size), determining if and when fish are added ("seeded" or "supplemented") to the stream, when females "carry" eggs and both when and where they spawn them, how movement occurs for each life stage (based on the strength of stream flow, fish density, and the time of year), how mortality and growth occur for each stage based on physical conditions (i.e., stream temperature and flow) and fish density. The order of rows specifies the sequence that each demographic process runs in the model. This is highlights the "realism" of the model – that the model reflects many of the biological processes that influence the numbers of fish in the system (but not all processes). In the first column, the demographic or accounting processes are specified in the order that they occur each time step. Functions describing the major demographic and accounting computations are described Function Definitions. Processes that operate on multiple specific stage and size classes are specified in the model by a suffix (underscore index - "_1", "_2"). The columns "start_year" and "end_year" specify the model years that each process in column is run. Columns "start_time" and "end_time" denote the index of the annual time steps when each process occurs; when "start_time" is 0 and "end_time" is 51, this corresponds to the process running the entire year from the first to the 52nd week of the year. Columns "start_comp_unit" and "end_comp_unit" denote, respectively, the zero-based indices of the valid upstream and downstream computational units for each process in column A. Columns "stage" and "class" denote the zero-based indices of the life stage and size classes; 0-7 correpsond to "stg_idx" in the species_info.txt. When processes operate on all stages, size classes, or both have 10 in the stage and classes columns.

After reading in the control_process.txt input file, the WEAPhish checks to ensure:

1. that spawning occurs before mortality occurs each time step to account for incubation related mortality (*in vivo* mortality of eggs, dewatering of redds, and redd superimposition)

2. that freshet and seasonal movement start and end times do not overlap within a year

FIGURE control_process.txt:

	A	B	C	D	E	F	G	H	I
1	process	start_year	end_year	start_time	end_time	start_comp_unit	end_comp_un	stage	classes
2	graduate	2010	2019	0	51	0	119	10	10
3	supplement	2010	2019	0	0	0	119	10	10
4	carry	2010	2019	15	22	0	119	10	10
5	spawn	2010	2019	15	22	0	119	10	10
6	freshet_movement_0	2010	2019	0	25	0	119	5	10
7	freshet_movement_1	2010	2019	0	25	0	119	6	10
8	freshet_movement_2	2010	2019	0	25	0	119	7	10
9	seasonal_movement_0	2010	2019	26	51	0	119	5	10
10	seasonal_movement_1	2010	2019	26	51	0	119	6	10
11	seasonal_movement_2	2010	2019	26	51	0	119	7	10
12	biomass	2010	2019	0	51	0	119	10	10
13	habitat_movement_0	2010	2019	0	51	0	119	0	10
14	habitat_movement_1	2010	2019	0	51	0	119	5	1
15	mortality	2010	2019	0	51	0	119	10	10
16	move_in_0	2010	2019	0	51	0	119	5	10
17	move_in_1	2010	2019	0	51	0	119	6	10
18	move_in_2	2010	2019	0	51	0	119	7	10
19	growth_0	2010	2019	0	51	0	119	4	10
20	growth_1	2010	2019	0	51	0	119	5	10
21	growth_2	2010	2019	0	51	0	119	6	10

spawn.txt

This input file contains information on the size of the “redd” (mean_area_per_redd), the range of suitable spawning temperatures, and the average weight of hatching salmon or fry (“embryo_wt”), and then yearly, “year”, details on the annual model time step when spawning occurs (“spawn_time_step”), the “fraction” of the total spawning in a given year that occurs in a given time step (i.e., the sum of these data over the spawning season for each year is 1.0), and any “notes” to help document this decision. Note that the model time step is defined based on when it makes sense biologically and physically to begin the model – this is the annual model time step. This file differs from its corresponding file in SALMOD, because the WEAPhish user can provide these details for each year rather than for one year (doing the copying, pasting, and editing in Excel might be easiest). For example, a user could specify how spawning times may change as with increasingly warm temperatures resulting from climate change.

FIGURE spawn.txt:

	A	B	C	D
1	species	chinook_salmon		
2	mean_area_per_redd	2.7	'avg. area per redd mound (n	
3	spawning_temp_range	2.2	18.9	'min-m
4	embryo_wt	0.245	'avg. wt(g) of embryos on em	
5	year	spawn_time_step	fraction	notes
6	2010	15	0.0075	.05 15-
7	2010	16	0.03375	.10 22-
8	2010	17	0.2075	.36 29-
9	2010	18	0.46375	.36 5-N
10	2010	19	0.2125	.10 12-
11	2010	20	0.0575	.03 19-
12	2010	21	0.015	none
13	2010	22	0.0025	none

supplement.txt

This input file is used to specify the number of fish (“num_fish”) that are added to in each “year” for each “stage” along the stretch of stream from the upstream, “seg_low”, to the downstream segment, “seg_high”. When WEAPhish reads these data, the “num_fish” are allocated based on the available habit in the corresponding stream segment defined from “seg_low” to “seg_high”. The available habitat depends on the fish stage, stream flow, and stream habitat type. The fish are assigned to be “weight” grams on average. The “sex_ratio” (fraction of females) is not used in WEAPhish and will be removed in future versions of the model. Currently, the model has only been implemented supplementing adult fish (i.e., returning spawners), but supplementing with juvenile fish (e.g., from a hatchery or wild fish from a tributary could be done, but will require testing and code modification).

FIGURE supplement.txt:

	A	B	C	D	E	F	G	H
1	stage	year	ann_timestep	seg_low	seg_high	num_fish	sex_ratio	weight
2	adult_female	2010	0	1	9	1559	0.5	7500
3	adult_female	2010	0	10	15	837	0.5	7500
4	adult_female	2010	0	16	24	1015	0.5	7500
5	adult_female	2010	0	25	36	1005	0.5	7500
6	adult_female	2010	0	37	54	985	0.5	7500
7	adult_female	2010	0	55	57	982	0.5	7500
8	adult_female	2010	0	58	74	540	0.5	7500
9	adult_female	2010	0	75	90	280	0.5	7500
10	adult_female	2010	0	91	105	182	0.5	7500
11	adult_female	2010	0	106	120	118	0.5	7500
12	adult_male	2010	0	1	9	1559	0.5	7500
13	adult_male	2010	0	10	15	837	0.5	7500
14	adult_male	2010	0	16	24	1015	0.5	7500
15	adult_male	2010	0	25	36	1005	0.5	7500
16	adult_male	2010	0	37	54	985	0.5	7500
17	adult_male	2010	0	55	57	982	0.5	7500
18	adult_male	2010	0	58	74	540	0.5	7500
19	adult_male	2010	0	75	90	280	0.5	7500
20	adult_male	2010	0	91	105	182	0.5	7500
21	adult_male	2010	0	106	120	118	0.5	7500

relations.txt

This file contains data on the functional relationships between the physical conditions and growth, mortality, movement, fecundity, as well as the biological link between weight and length and weight to eggs.

- movement_freshet: this section describes the distance moved downstream when young fish experience anomalously high flows – a freshet. When a freshet occurs, each “stage” and “class” moves downstream “distance_moved” meters. Only the fraction “proportion_moved” moves and those moving experience a mortality rate equal to “proportion_dying”. Freshets are defined as either: twice the flow of the previous time step **OR** greater than or equal to twice the average flow of the three previous time steps [2xPorA] *or* twice the flow of the previous time step **AND** greater than or equal to twice the average flow of the three previous time steps [2xPandA]. Note: In WEAPhish, like Salmod, only juvenile fish experience freshet movement.

FIGURE freshet_movement:

movement_freshet				
stage	classes	distance_mov	proportion_moved	proportion_dying
fry	1	15000	0.5	0.3
fry	2	15000	0.5	0.3
fingerling	1	15000	0.5	0.2
fingerling	2	15000	0.5	0.15
juvenile	1	15000	0.5	0.05
juvenile	2	15000	0.5	0.03
juvenile	3	15000	0.5	0.01

- movement_seasonal: this section describes the distance moved downstream when young fish make the ontogenetic migration to the ocean. When seasonal movement occurs during a time step, each “stage” and “classes” moves downstream “distance” meters, but only the fraction “proportion” moves, and those moving experience “mortality”. Note: In WEAPhish, like Salmod, only juvenile fish undergo seasonal movement (i.e., migration). Also, freshet and seasonal movement may not occur during the same time step.

FIGURE movement_seasonal:

11	movement_seasonal					
12	stage	classes	time	distance	proportion	mortality
13	fry	1	26	1500	0.6	0.01
14	fry	1	27	1500	0.6	0.01
15	fry	1	28	1500	0.6	0.01
16	fry	1	29	1500	0.6	0.01
17	fry	1	30	1500	0.6	0.01
18	fry	1	31	1500	0.6	0.01
19	fry	1	32	1500	0.6	0.01
20	fry	1	33	1500	0.6	0.01
21	fry	1	34	1500	0.6	0.01
22	fry	1	35	1500	0.6	0.01
23	fry	1	36	1500	0.6	0.01

- growth_temperature: this section describes the how much each “stage” will “grow” based on the “temperature” in Celcius. In the model, the actual growth at a given time step is based on a linear interpolation of the relationship described by the growth_temperature relationship data for each stage. Note: negative growth

in terms of fish weight can occur at high temperatures, but fish do not shrink in length.

FIGURE growth_temperature:

151	growth_temperature		
152	stage	temperature	daily_growth
153	eggs	1	0.0025
154	eggs	2	0.0031
155	eggs	3	0.0038
156	eggs	4	0.0046
157	eggs	5	0.0055
158	eggs	6	0.0065
159	eggs	7	0.0076
160	eggs	8	0.0088

- mortality_base: this section specifies the “base_mortality” or background natural mortality experienced by each “stage” during a single time step.

204	stage	base_mortality
205	eggs	0.035
206	fry	0.025
207	fingerling	0.025
208	juvenile	0.025
209	adult_female	0.003
210	adult_male	0.003
211	spawning_female	0.003
212	spawning_male	0.003

- mortality_temperature: this section specifies the “mortality” that each “stage” experiences each base on stream “temperature”. In the model, the actual mortality incurred at a given time is based on a linear interpolation of the mortality rate corresponding to the stage specific temperature relationship described by the mortality_temperature data.

FIGURE mortality_temperature:

213	mortality_temperature		
214	stage	temperature	mortality
215	eggs	2	0.0426
216	eggs	5	0.0068
217	eggs	8	0.0064
218	eggs	11	0.0115
219	eggs	14	0.0827
220	eggs	40	1
221	fry	0	0
222	fry	15.5	0
223	fry	20	0.175
224	fry	25	0.8

- weight_length: this section specifies the length weight relationship, where a fish of “length” mm has “weight” grams. This relationship is used to interpolate lengths (weights) from weights (lengths). Note: this relationship is $y \sim ax^b$, where $b \sim 3$.

FIGURE weight_length:

340	weight_length	
341	length	weight
342	20	0.08773202
343	30	0.29489744
344	40	0.69700809
345	50	1.35830955
346	60	2.34288344
347	70	3.7146815
348	80	5.53754801
349	90	7.87523545
350	100	10.7914162
351	110	14.3496918

- mortality_movement: this section describes the stage species mortality rate that corresponds to the distance moved resulting from density dependent habitat

movement. Each “stage” and “classes” incurs a “mortality” penalty for the “distance_moved”.

FIGURE mortality_movement:

372	mortality_movement			
373	stage	classes	distance_mov	mortality
374	fry	1	0	0
375	fry	1	16000	0
376	fry	1	17000	1
377	fry	2	0	0
378	fry	2	16000	0
379	fry	2	17000	1
380	fingerling	1	0	0
381	fingerling	1	3000	1
382	fingerling	2	0	0
383	fingerling	2	3000	1

- mortality_density: For each “stage” the fish model finds the “mortality” rate associated with the “density” of fish based on a linear interpolation of the relationship described in the mortality_density data. There must be at least two rows for each stage in order to provide a lower and an upper density that can be used to establish a relationship used for linear interpolation. In the example, there is no density dependent mortality – mortality is zero at the upper and lower density bounds for each stage.

FIGURE mortality_density:

390	mortality_density		
391	stage	density	mortality
392	adult_female	0	0
393	adult_female	3	0
394	adult_male	0	0
395	adult_male	3	0
396	spawning_fem	0	0
397	spawning_fem	3	0
398	spawning_mal	0	0
399	spawning_mal	3	0
400	eggs	0	0
401	eggs	1000	0
402	fry	0	0

- weight_egg: this section describe the weight fecundity relationship for the study species. A female fish with “weight” grams yields “eggs” many eggs. Fecundity of a fish of “weight” is linearly interpolated from these data.

FIGURE weight_egg:

408	weight_egg	
409	weight	eggs
410	856	0
411	1718	889
412	3035	2426.8
413	4910	3964.6
414	7449	5502.4
415	10757	7040.2
416	14943	8578
417	20118	10115.8
418	26394	11653.6
419	33881	13191.4
420	42695	14729.2
421	52949	16267

wua.txt

This file defined the WUA or weighted usable area (units ft^2/ft) for each stage and habitat type. The first row lists the species. The second row lists the “stages_with_wua” beginning in the second column. The third row contains “stages_scalar” or the stages without explicit WUA data (see the next entry “wua_stage_scalars.txt” for more information). The WUA data begin on the fourth row. The “index” column tracks the

number of data flow:wua data pairs for each (meso-)habitat type (“meso_type”) and life “stage”, where “flow” determines the available “wua”. For each “stage” the fish model finds the “wua” associated with the “flow” based on a linear interpolation of this empirical relationship.

FIGURE wua.txt

	A	B	C	D	E
1	species	chinook_salmon			
2	stages_with_wua	fry	fingerling	adult_female	spawning_female
3	stages_scalar	eggs	juvenile	adult_male	spawning_male
4	index	meso_type	stage	flow	wua
5	1	P	fry	0	0
6	2	P	fry	20	17.758
7	3	P	fry	30	17.355
8	4	P	fry	40	16.498
9	5	P	fry	50	15.558
10	6	P	fry	60	14.708
11	7	P	fry	70	13.841
12	8	P	fry	80	13.142

wua_stage_scalars.txt

This is a “convenience” file to reduce redundant data entry when WUA data for one stage applies to two or more life stages or the WUA data for one given stage is a simple multiplier of another. The first column is the “ref_stage” or reference stage that has WUA defined in the wua.txt file, the “scale_stage” column list the stage that has the same WUA as the reference data, and the “multiplier” indicates how the WUA data for the “ref_stage” scales to the “scale_stage”. Here, all the “scale_stages” have the same WUA as the “ref_stage” so the “multiplier” is 1.0. You can also see that when WUA data for one “ref_stage” applies to more than one “scale_stage”, the relationship for each “scale_stage” is entered on a separate line.

FIGURE wua_stage_scalars.txt

	A	B	C
1	ref_stage	scale_stage	multiplier
2	fingerling	juvenile	1
3	adult_female	adult_male	1
4	spawning_fem	spawning_mal	1
5	spawning_fem	eggs	1

stream.txt

This file contains information on the spatial configuration of the stream system. The first row contains the number of stream habitat units (120 in the figure below). Beginning in the second row, which corresponds to the most upstream segment, the first column contains the length of stream segment length and the second column contains the habitat type (e.g., “P” for Pool, “RU” for Run, and “RF” for Riffle), and the third column contains notes. Habitat types in this file must correspond to the “meso_type” in wua.txt (meso-habitat type).

FIGURE stream.txt: number of computational units and beginning of stream computational unit length and habitat type data.

	A	B	C
1	120	NumberCUs	CDFG has reac
2	264	P	A1. Reach A, 5
3	238	RU	Three habitat t
4	142	RF	Use this file fo
5	284	P	A2
6	256	RU	
7	152	RF	
8	185	P	A3
9	167	RU	
10	99	RF	

The next set of information in stream.txt maps how available stream temperature and flow data map to the stream computational units in the study area. This is because, in many instances, flow and temperature data will not be available for each stream computational unit defined in a WEAPhish model and, therefore, single flow and temperature data value are applied to a range of computational units. In the figure below, row 122 says that there are 8 “flow segments”, which means that flow data are available for 8 locations along the stream that get mapped to the 120 stream habitat or “computational” units as defined by the segment indices in row 123 to 130 in the figure below (the stream computational indices are 1, 16, 40, 55, 58, 73, 97, 115). This means that the first column of weekly (or other time step) *flow* data in flows.txt is applied from river segment 1 (most upstream) to segment 15, the second column applies from segments 16 to 39, …, and the eighth column of flow data applies from segments 115 to downstream boundary, which, in this example, is 120. The temperature data are mapped to the stream computational units in the same way. Flow and temperature data that come directly from WEAP (or possibly another hydrological model) will likely be

available for each computational unit. In these instances, the computational unit indices run from 1, 2, 3, …, *n-1*, where *n* is the number of stream computational units.

FIGURE stream.txt: Flow and stream segment information.

122	8	Number of flow segments	
123	1		
124	16		
125	40		
126	55		
127	58		
128	73		
129	97		
130	115		END of flow se
131	23	Number of temperature segm	
132	1		
133	5		
134	9		
135	13		
136	16		
137	20		
138	24		

The final component of stream.txt describes the how fish view the extreme upstream (Gate1) and downstream (Gate2) segments. These may be seen as:

- "Barrier" = 4: no movement across this boundary
- "Downstream" = 1: only downstream movement is allowed
- "Upstream" = 2: only upstream movement is allowed
- "Bi-directional" = 3: allows both upstream and downstream movement across boundary

FIGURE stream.txt: End of temperature segement data and upstream boundary (Gate1) and downstream boundary (Gate2) rules. Here, the upstream gate, Gate1, is a barrier to movement in both directions, while the downstream gate, Gate2, allows movement downstream.

154	115		END of temper
155	Gate1	4	Barrier
156	Gate2	1	Downstream

temps.txt

This file contains temperature (°C) data for each “temperature segment” in the columns for all time steps (rows). The first column of this file could contain the date, but it is not necessary (date data should have the format “MM-DD-YYYY”).

FIGURE temps.txt:

	A	B	C	D	E	F	G
1	15.23	15.47	15.64	15.81	15.92	16.01	16.13
2	17.83	18.26	18.57	18.88	18.95	19.12	19.32
3	18.85	19.34	19.69	20.04	20.1	20.29	20.52
4	16.83	17.11	17.29	17.47	17.55	17.65	17.76
5	11.93	12.09	12.21	12.33	12.38	12.4	12.48
6	14.82	15.04	15.2	15.38	15.47	15.56	15.67
7	15.99	16.27	16.46	16.64	16.75	16.86	16.97
8	18.4	18.8	19.08	19.37	19.44	19.59	19.77
9	18.79	19.26	19.6	19.95	20.03	20.21	20.44

flows.txt

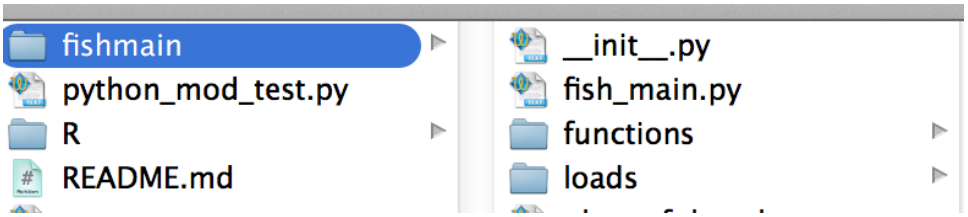
This file contains flow (cfs) data for each “flow segment” in the columns for all time steps (rows). The first column of this file could contain the date, but it is not necessary (date data should have the format “MM-DD-YYYY”).

FIGURE flows.txt:

	A	B	C	D	E	F	G	H
1	86.95	89.86	95.75	278.75	278.75	282.38	282.38	307.36
2	65.28	67.87	73.25	256.25	256.25	259.41	259.41	281.92
3	60	62.35	67.29	242.12	242.12	244.98	244.98	265.56
4	60	62.19	66.76	238.26	238.26	240.92	240.92	260.03
5	251.59	253.75	258.21	441.21	441.21	443.88	443.88	462.9
6	121.37	123.49	127.83	310.83	310.83	313.46	313.46	331.93
7	64.51	66.49	70.54	253.54	253.54	255.99	255.99	273.24
8	60	61.87	65.67	232.79	232.79	235.1	235.1	251.31

WEAPhish code files and simulation

Once the model input data files have been created and saved into the selected directory for the simulation, the next step is to run the model. Before launching the model, the code structure is discussed briefly so the user will understand where to look if (when) runtime errors occur. The WEAPhish code files reside in a directory named “fishmain”. “(F)ishmain” contains the file “fish_main.py” and the directories “functions” and “loads”. “(F)ish_main.py” is the script that is called to run the specified simulation. “(F)unctions” contains the file “functions.py” that contains the definition of all functions defined for running WEAPhish and an empty “__init__.py” file, which indicates that the directory is a Python “module” that can be loaded with an `import` statement. The “loads” directory contains “loads.py”, which reads all input data and prepares the model for the simulation, and an empty “__init__.py” file. The screenshot below shows the fishmain directory and its contents.



WEAPhish is written as a Python script and is currently implemented by command line calls to the Python interpreter to run the main WEAPhish script fish_main.py. *It is not a stand alone executable at this time*. The input data files described above should all be in the same directory, which has a descriptive name for the model run (e.g., or perhaps with a nested file structure: river_name/hydrological_model_version/species/low_mortality/). Before telling Python to run fish_main.py, change the working directory to the one containing the input data. Then launch the fish_main.py script from the command line (Python’s script mode).

On Windows:

```
> chdir c:\path\to\input\files
```

or

```
> cd c:\path\to\input\files
```

then

```
> python c:\path\to\fishmain\fish_main.py
```

And on OSX or Linux:

```
> cd ~/path/to/input/files
```

```
> python ~/path/to/fishmain/fish_main.py
```

which runs the model using the in input data files in the working directory. Note: If Ipython (the interactive Python shell) is available on your computer, running “ipython path/to/fish_model/fish_main.py” from the command line will provide, among other things, color coded syntax highlighting that is useful when debugging model runs. When called from the command line, WEAPhish takes about 20 minutes to run for a one year simulation on a stream with 120 computational units (on a 2 GHz Intel Core i7 MacBook Pro with 16 GB RAM).

The figure below describes the WEAPhish simulation process:

Beginning of simulation:

Load input data, set model parameters, initialize data structures, define simulation type and time range

Each simulation year:

Seed the stream with fish, based on their observed distribution and habitat use

Each time step:

Each process at stream computational unit level:

Compute biomass, add additional fish (e.g., hatchery supplementation), spawn (during spawning season), and return moving fish from the “virtual stream” to new their stream computational unit

Each cohort of fish within a computational unit:

Calculate mortality of juveniles and adults based on temperature, density, and base rates, calculate growth and determine new size class, determine seasonal, flow (i.e., freshet), and density dependent movement and mortality.

Tabulate simulation results

State-variable Arrays

Like Salmod, WEAPhish tracks the current system "state"[1] by updating a set of two-dimensional arrays that contain the current information on the number of fish, life stage, size class, mean weight, mean length, and mean number of eggs per female for each group in a stream computational unit. The number of rows in each array is equal to the number of stream habitat units (i.e., computational units) and the number of columns is equal to the maximum number of groups. For eggs and juvenile life stages "groups" correspond to individual cohorts born in a given time step and stream habitat unit. There may be multiple groups for each of these stages because spawning typically occurs over multiple weeks in the model and each cohort spawned in a given week would be assigned to a new group (a column index in each array). Movement of juvenile fish from one computational unit to the next will generate a new "group" in the new computational unit. If the number of groups in a stream habitat unit exceeds the maximum number of groups allowed, the WEAPhish finds two groups of the same life stage and size class to merge into one group. After merging, the "new group" column index becomes that corresponding to the final column of the state arrays. The next new group will be assigned to that column index in the corresponding stream habitat unit. When there is more than one stage and size class group combination with two or more groups present, WEAPhish randomly selects one pair of groups to minimize selection bias of one stage/size class. Adult and spawning stages have only one group and they are always assigned to the first 4 columns: adult females in first column, adult males in second column, spawning females in third column, and spawning males in the fourth column. These structures are invisible to the user, but are critical for WEAPhish functioning.

number_a: the current number (integer) of fish or eggs in the specified computational unit and group. 0 (zero) indicates no fish are in a computational unit/group array element.

stage_a: the integer index of the current stage of fish in the specified computational unit and group. 0 = adult female, 1 = adult male, 2 = spawning female, 3 = spawning male, 4

---

[1] By "state" of the system I mean that we track the information needed to project the population forward one time step.

= eggs, 5 = fry, 6 = fingerling, 7 = juvenile. -999 corresponds to no fish in a computational unit/group.

class_a: integer index of the size class for the corresponding life stage in a computational unit/group. Ranges from (1 to 5, the maximum number of size classes per stage) and 0 indicates an empty array element.

weight_a: contains mean weight of fish in a computational unit/group. For the egg stage, weight_a contains the relative maturity of eggs in a computational unit/group (0-100%).

length_a: contains mean length of fish in a computational unit/group. For the egg stage, length_a contains, current minimum of habitat area (in terms of weighted usable area) for incubating eggs.

repro_factor_a: contains the mean number of eggs per female in a computational unit/group. For the egg stage, repro_factor_a contains the number of redds in a cohort.

stage_in_cu_a: this array tracks the presence of each stage in a computational unit during the current time step. It has the dimensions (number of rows = number of computational units, number of columns = number of stages, 8).

Function Definitions

Fish model function definitions

continuous_sim(): This function is called each model time step and provides the structure for WEAPhish simulations.

move_in(): After fish have moved during a time step, this function returns all moving fish to their “new” stream computational unit.

spawn(): This function controls when and what fraction of fish spawn during the breeding season.

compute_biomass(): This function calculates the summary abundance data

cohort_process(): Calls the appropriate process that operates on individual cohorts within a computational unit (i.e., mortality, habitat_movement, seasonal_movement, freshet_movement, growth, graduate).

mortality(): Computes the base, temperature, and density related mortality for fish groups. In addition, *in vivo* egg mortality, egg superimposition, and egg mortality due to changes in flow leading to “dewatering” of eggs.

habitat_movement(): Moves fish if density exceeds the capacity of the computational unit for the given stage.

seasonal_movement(): Moves fish downstream during the out-migration season. Cannot occur during the same time steps as freshet_movement().

freshet_movement(): Moves fish when anomolously high flows occur during a time step.

growth(): Computes the temperature stage-based growth for each cohort of fish.

graduate(): Determines if a cohort of fish has reached a new size class and modifies the stage and class accordingly.

Excel “middle-ware” for running WEAP and WEAPhish

While WEAPhish will work as a stand alone tool to evaluate ecological impacts of stream flow and temperature on fish for any stream with sufficient data, the value of a model like WEAPhish increases when used in conjunction with tools that expand the scope of questions investigated to include the trade-offs involved analyzing the costs and benefits of proposed water resource operational strategies. For WEAP users interested in using WEAPhish, “middle-ware” called Master_WEAPhish.xlsm scripted Visual Basic for Applications (VBA) in Excel provides and convenient interface to both models. Contact Samuel Sandoval (samsandoval@ucdavis.edu) or Sooyeon Yi (scyi@ucdavis.edu) for the Master_WEAPhish.xlsm macro-enabled workbook and user’s manual.

Output Data Files

All output data are stored as text (*.txt) files. WEAPhish currently produces two sets of output data. The first is termed “derived” data and these data describe of physical conditions and habitat-based capacity of the stream system over space and time. These files include the full temperature and flow data sets for each time step and computational unit, as well as, stage specific weighted usable area, weighted usable area multiplied by stream length (“total” usable area or TUA), and carrying capacity.

Derived Data

File names: out_flows.txt and out_temp.txt

Flow and temperature data are stored in the ‘derived’ directory (i.e., ‘derived/out_flows.txt’ and ‘derived/out_temp.txt’). These files contain the flow and temperature for each week of the model run with the date in the first column and the flow or temperature in columns 2 to the number of stream computational units plus 1. Data in column 2 corresponds to the most upstream computational unit, and the last column (number of stream computational unit plus 1) holds data for the most downstream computational unit. The stream (meso-)habitat type (e.g., **P** – pool, **RU** – run, or **RF** – riffle) for each computation are stored in the first row, from columns 2 to number of stream computational unit plus 1.

date	**P**	**RU**	**RF**	**…**
1/1/2014	20	21	21.2	21.5
1/8/2014	23	24	25.2	25.5

Carrying Capacity and WUA Table

File names: WUA_df_i.txt, TUA_df_i.txt, and K_df_i.txt, where i = 1, 2, …, 7 and corresponds to the index for each life stage. That means there are eight separate files for each data type.

Flow influences the available habitat, which determines the “carrying capacity” of each stream segment (i.e., computational unit) in terms of the numbers or biomass that can be supported for each stage of fish for each model time step. If the carrying capacity is exceeded, then fish must move until they find suitable habitat (the movers incur mortality based on how far they move before finding suitable habitat). WUA or weighted usable area is a function of the life stage, flow, and habitat type. What I've termed “TUA”, total usable area in a segment (m^2), is the product WUA and the length of the stream segment. Carrying capacity is the stage specific density value (numbers and biomass) per area (m^{-2}) time TUA. The data in each file have the same structure as the flow and temperature data and are also stored in the ‘derived’ directory (e.g., ‘derived/WUA_df_1.txt’).

date	P	RU	RF	…
1/1/2014	20	21	21.2	21.5
1/8/2014	23	24	25.2	25.5

Output Data

System State Log

File name: system_state_log.txt

This contains the primary results of the simulation: fish abundance over time, space, and stage. WEAPhish stores abundance (both in numbers and biomass) data for each stage, time step, and stream segment. WEAPhish also stores mean weight, length, and, for adult/spawning females, eggs per female at the same resolution (these are the main state variables of WEAPhish). The system state data in ‘system_state_log.txt’ look like:

date	annual_time_step	comp_unit	stage	stg_idx	class	number	biomass	mean_wt	mean_len	repro_factor
1/1/2014	0	0	adult_female	0	0	245	539000.0	2200.0	950.0	4000.0
1/1/2014	0	0	spawning_female	1	0	30	70500.0	2350.0	940.0	3900.0

but **without the header row**. The data are stored in the ‘out’ directory (‘out/system_state_log.txt’). Note that for the egg stage, weight data correspond to the fractional maturation of eggs, length data correspond to effective incubation area in the stream segment, and the eggs per female correspond to the number of redds in composing the cohort.

Dates and Time Step Reference Table

File name: date_time_step_ref.txt

This table contains the date of each model time step and the corresponding total time step (which runs from 0, 1, 2, …, *N-1*, where *N* is then total number of model time steps (because Python counts beginning with 0), annual time step (which runs from 0, 1, 2, …, *n*, where *n* is the number of time steps in a model year less one, the number of weekly time steps in a year may not always be 52), and model year (which runs from 0, 1, 2, …, *y*, where *y* is the number of study years). This table will be helpful when summarizing, joining, or performing other data wrangling with data tables that store different time tracking variables. The file is written to the ‘out’ directory (‘out/ date_time_step_ref.txt’). While the table below starts on the new calendar year, the biological/model year will, in most applications, begin at a different time in the year.

date	total time step	annual time step	model year
1/1/2014	0	0	0
1/8/2014	1	1	0
1/15/2014	2	2	0
1/22/2014	3	3	0
1/29/2014	4	4	0

Mortality Statistics

File name: mort_stats.txt

This file records each mortality event (the mortality ‘rate” and number of ‘deaths’) at the group level by cause. The file is written to the ‘out’ directory (‘out/ mort_stats.txt’).

time_step	stage	stage_idx	class	comp_unit	group	cause	rate	deaths
0	adult_male	1	0	0	0	base	0.02	40
0	adult_male	1	0	0	0	temperature	0.1	200
0	adult_male	1	0	0	0	density	0.05	10

Movement Statistics

File name: move_stats.txt

This file records each movement event (the number moving successfully and number dying and corresponding distance moved) at the group level by cause. The original and new stream computational units are also recorded. The file is written to the ‘out’ directory (‘out/ move_stats.txt’).

time_step	stage	stage_idx	class	comp_unit	new_comp_unit	group	cause	deaths	average_dying_distance	numbers	average_mover_distance
0	adult_male	1	0	0	4	0	seasonal	40	1500	1000	1500
0	adult_male	1	0	0	4	0	seasonal	200	1500	1000	1500
0	adult_male	1	0	0	4	0	seasonal	10	1500	1000	1500

Initial Fish Seeding Data (Supplementing):

File name: supp.txt

This file contains the columns:

ann_timestep: the time step within the current model year

stage: name of the “seeded/supplemented” stage

year: calendar year corresponded to current model year

comp_unit: computational unit

dist_num: the distributed number of fish “seeded” into the computational unit based on available habitat

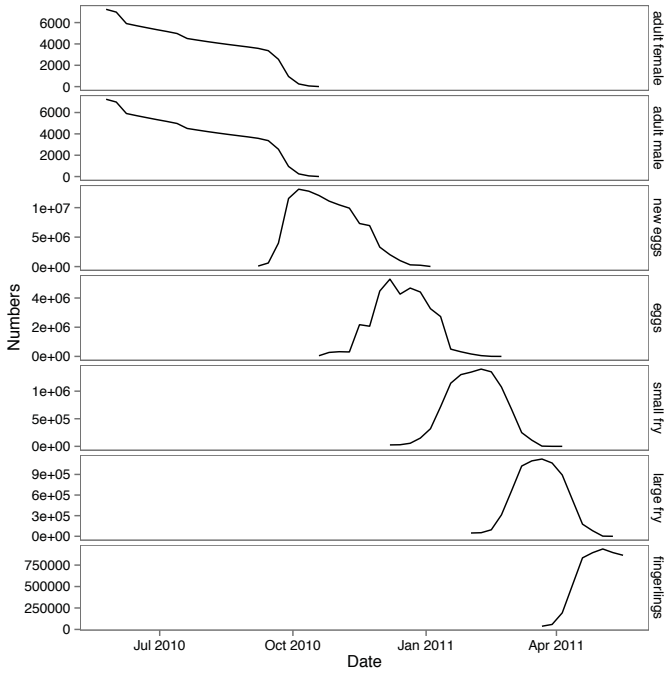
It is stored in the ‘out’ directory (‘out/supp.txt’).

ann_timestep	stage	year	comp_unit	dist_num
0	adult_female	2010	1	492
0	adult_female	2010	2	43
0	adult_female	2010	3	25

Data visualization

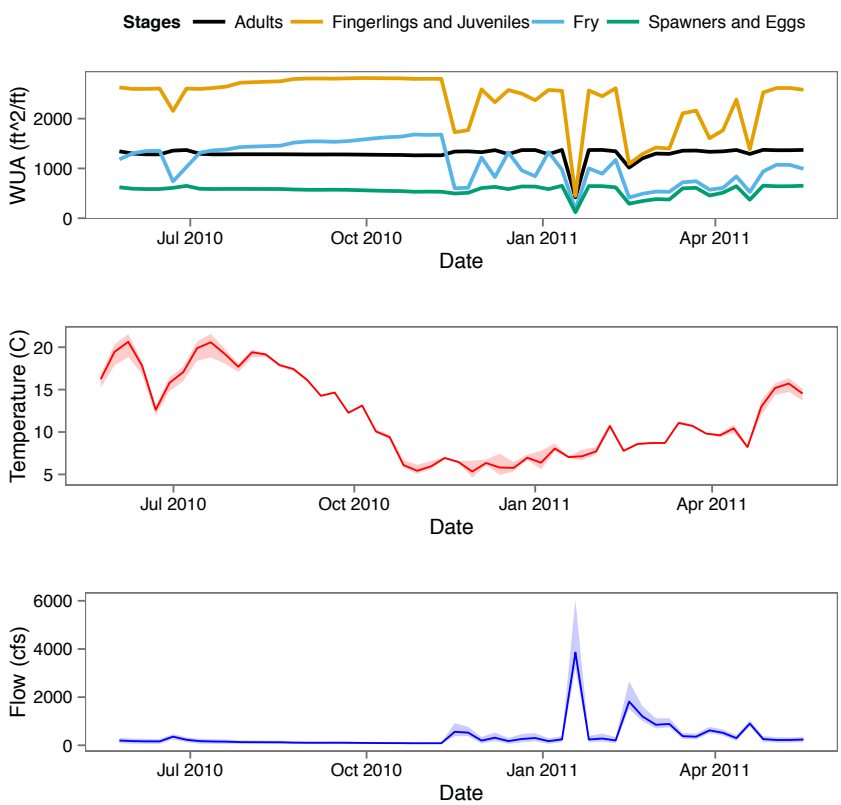
Because output data are stored in text (*.txt) files, users have the option to use the software of their choice to visualize the data. Below are a few example plots of output data produced using open-source software R (www.r-project.org) and the open source IDE RStudio (www.rstudio.com) that summarize output of different example WEAPhish runs.

The figure below shows a time series plot of fish abundance (Numbers) over time (one model year from May 2010 to May 2011) for each stage. This plot uses the ggplot2 R package (http://ggplot2.org/) and employs the 'faceting' approach to produce a 'stack' of time series plots for each stage in a single column. These plots show the time evolution from one generation to the next – adult males and females in the top two plots spawn and die in September and early October, but produce new eggs – as well as the growth based transition from one stage to the next (new eggs grow into eggs, which grow into small fry, etc.). Note that the y-axis varies for each of the plots.

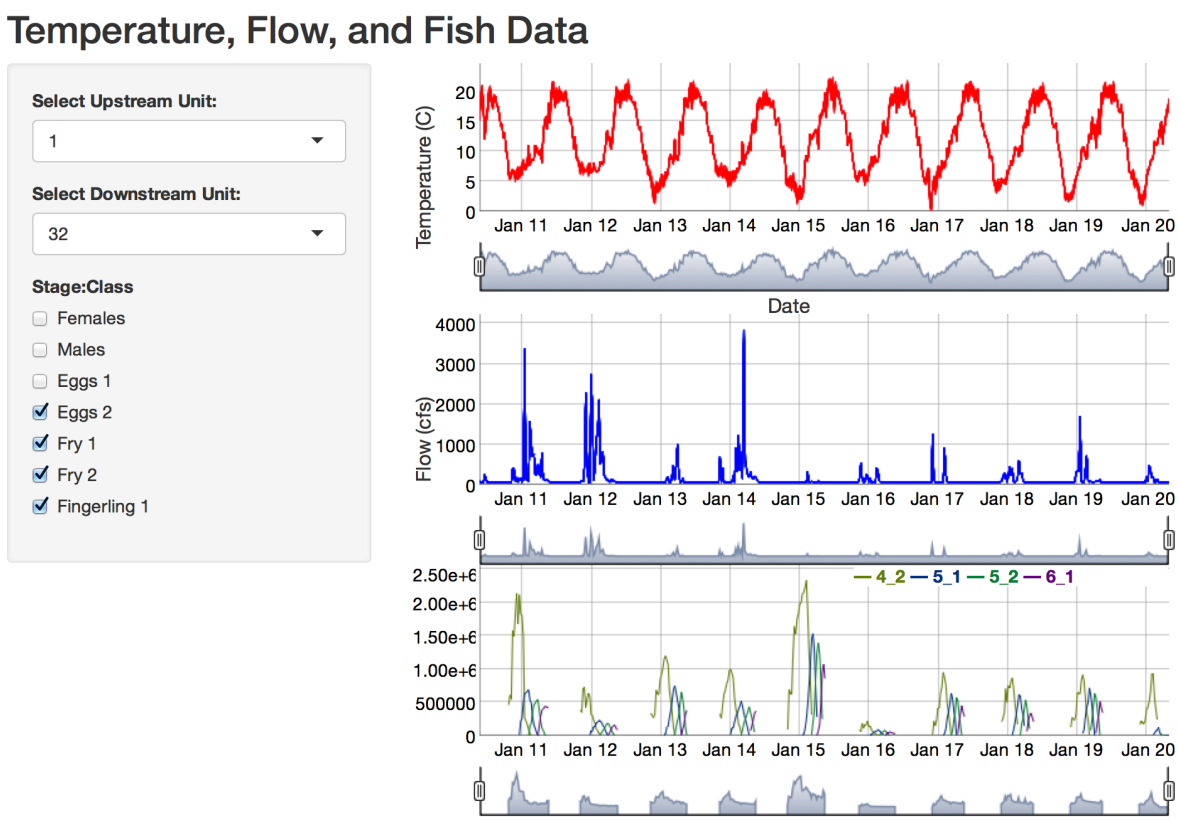


The following plot shows an example of how the physical habitat changes with time and how that impacts the available fish habitat. Each plot below displays one model year of data (May 2010 to May 2011). The top plot shows changes in weighted usable area (WUA) for all stages, the middle and bottom plots show the mean and min/max temperature (red) and flow (blue) data.

In early 2011, we can see that a high flow event (spike in bottom plot) resulted in a temporary reduction in usable stream habitat for all stages in the top plot (a dip in all lines). From the previous plot, we see that this corresponds to a realized loss of habitat only for egg and fry stages, as the other stages are not present in the stream.



Another plotting option is to develop interactive graphics that allow the user to actively investigate their data without having to edit code. Below is a snapshot of a plot developed using RStudio's Shiny web-application framework (http://cran.r-project.org/web/packages/shiny/index.html) and also uses the dygraphs package for creating interactive graphics (http://rstudio.github.io/dygraphs/index.html). This plot displays temperature (top, red), flow (middle, blue), and fish abundance (bottom) data from a ten-year example simulation. The Select Upstream/Downstream Unit tabs on the left allow the user to specify the computational units to display for the temperature and flow data (here, computational units 1 to 32 are shown). The Stage:Class allows the user to plot time series of the selected life stage (here, eggs size class 2, fry size classes 1 and 2, and fingerlings size class are plotted). At this course scale we can see that there is interannual variation in egg and juvenile production, but the finer scale details of events that caused the variability are difficult to see for any given year. The slider bars below each time series plot allow the user to "drill-down" to finer time scales so that a shorter period of interest is displayed on each plot.



Bibliography

Bartholow, J., J. Heasley, J. Laake, J. Sandelin, B.A.K. Coughlin, A. Moos, and Johnson Controls World Services, Inc. 2002. SALMOD: A population model for salmonids: User's manual. Version W3. U.S. Geological Survey, Fort Collins, CO.

Bartholow, J., J. Laake, C.B. Stalnaker, and S. Williamson. 1993. A salmonid population model with emphasis on habitat limitations. Rivers, 4: 265-279.

Thompson, L. C., M. I. Escobar, C. M. Mosser, D. R. Purkey, D. Yates, and P. B. Moyle. 2012. Water management adaptations to prevent loss of spring-run Chinook salmon in California under climate change. Journal of Water Resources Planning and Management. 138: 465–478.