Grand Valley State University
Robert B. Annis Water Resources Institute

A New Computer Program to Evaluate Biases in the Two-Sample Mark-Recapture Abundance Estimator

Christian Yap*, Carl R. Ruetz III*[1], and James McNair*[1]

[1]Annis Water Resources Institute, Grand Valley State University, Muskegon, Michigan

*E-mail: yap@mail.gvsu.edu, ruetzc@gvsu.edu, mcnairja@gvsu.edu

Abstract

Two-sample mark-recapture sampling is a common method used to estimate fish abundance. The idea is to capture and mark fish in an initial sample. The fish are then released to mix randomly with the whole population. A second sample is obtained, and the number of marked and unmarked fish is recorded. The Chapman estimator uses the number of fish marked in the first sample, the total number of fish captured in the second sample, and the number of recaptured fish to estimate abundance. The assumptions are: 1) the population is closed, meaning no immigration, emigration, births, or deaths, 2) all fish are equally vulnerable to being captured during each sample, meaning marking does not change the behavior of fish, and 3) marks are not lost or overlooked. Violations of these assumptions can happen frequently and examining bias when a combination of assumptions are violated is difficult. To explore how simultaneously violating multiple assumptions of the Chapman estimator affects bias, we developed a computer simulation using python software that allows the end user to assess bias by simulating a closed or open population, varying fish capture probabilities, and allowing fish to lose marks. The resulting simulations allow the end user to violate any combination of model assumptions and decide the magnitude of such violations to provide an estimate of bias. This software should be useful to fisheries managers that use the Chapman estimator or instructors that teach the two-pass mark-recapture sampling for abundance estimation.

Introduction

- 2-sample mark-recapture is a commonly used method to estimate fish population size.
- For outcomes to have the most accurate result, the method must meet certain assumptions.
- Violating those assumptions can result in bias, and assessing the resulting bias is often difficult.
- This computer software was developed to help users assess different biases when the violations do occur.

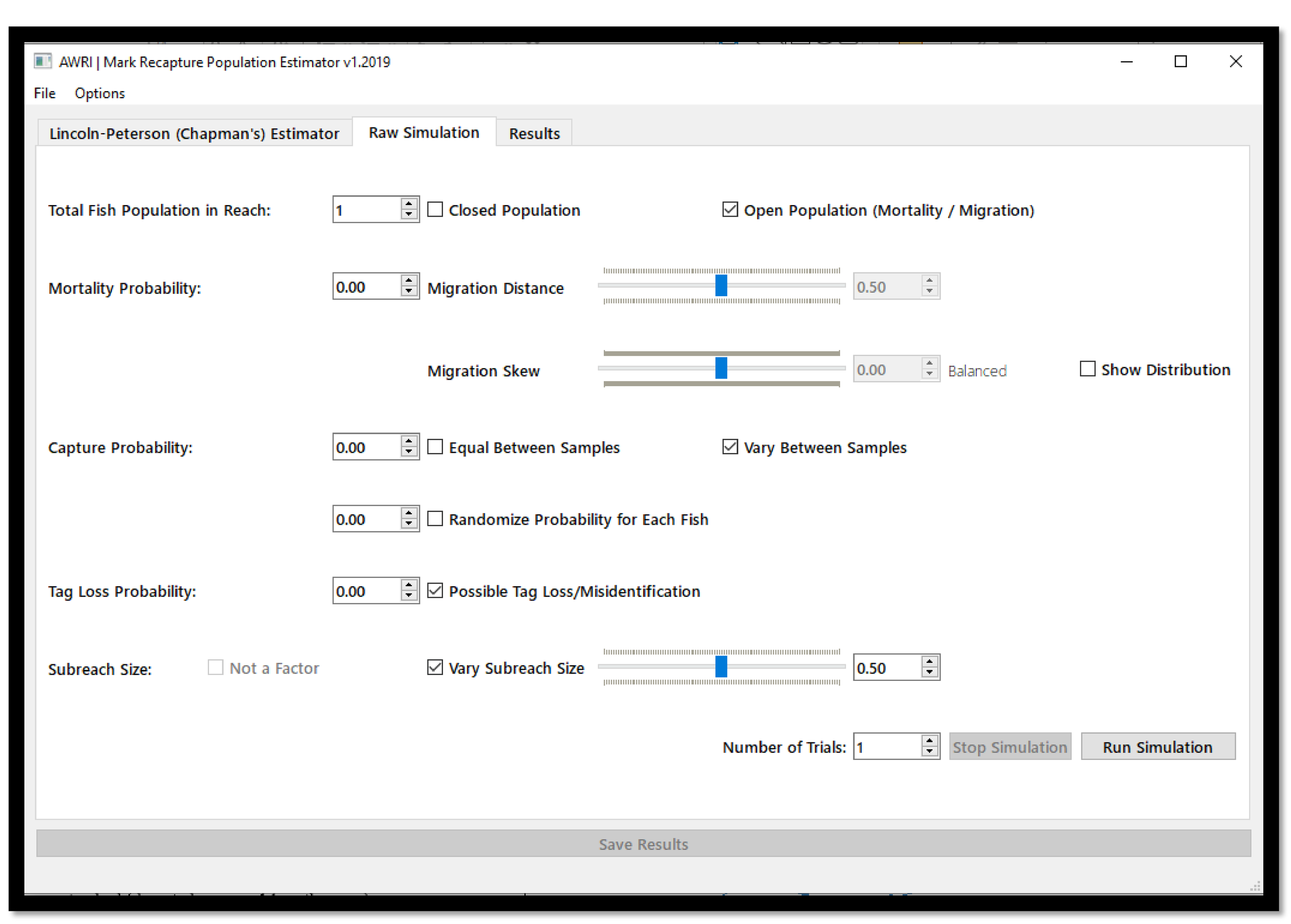

Fig. 1. Main graphical user window when the user starts up the software.

Methods

Upon start-up of software, a user can manipulate the following variables under the "Raw Simulation tab":

➢ **Total Fish Population in Study Reach**

➢ **Closed or Open Population** – With closed population, both mortality and fish migration in and out of the study reach are not occurring. *See figure 2.*

➢ **Capture Probability** – The chance of capturing a fish. This can be equal between all sampling or varied. It can also be completely random for each fish.

➢ **Tag Loss & Misidentification Probability** – The chance of a fish losing its tag or a user not seeing the tag.

- **Sub-reach Size:** The percent of the study area being sampled.
- **Number of Trials:** This is the total number of times to repeat the simulation.

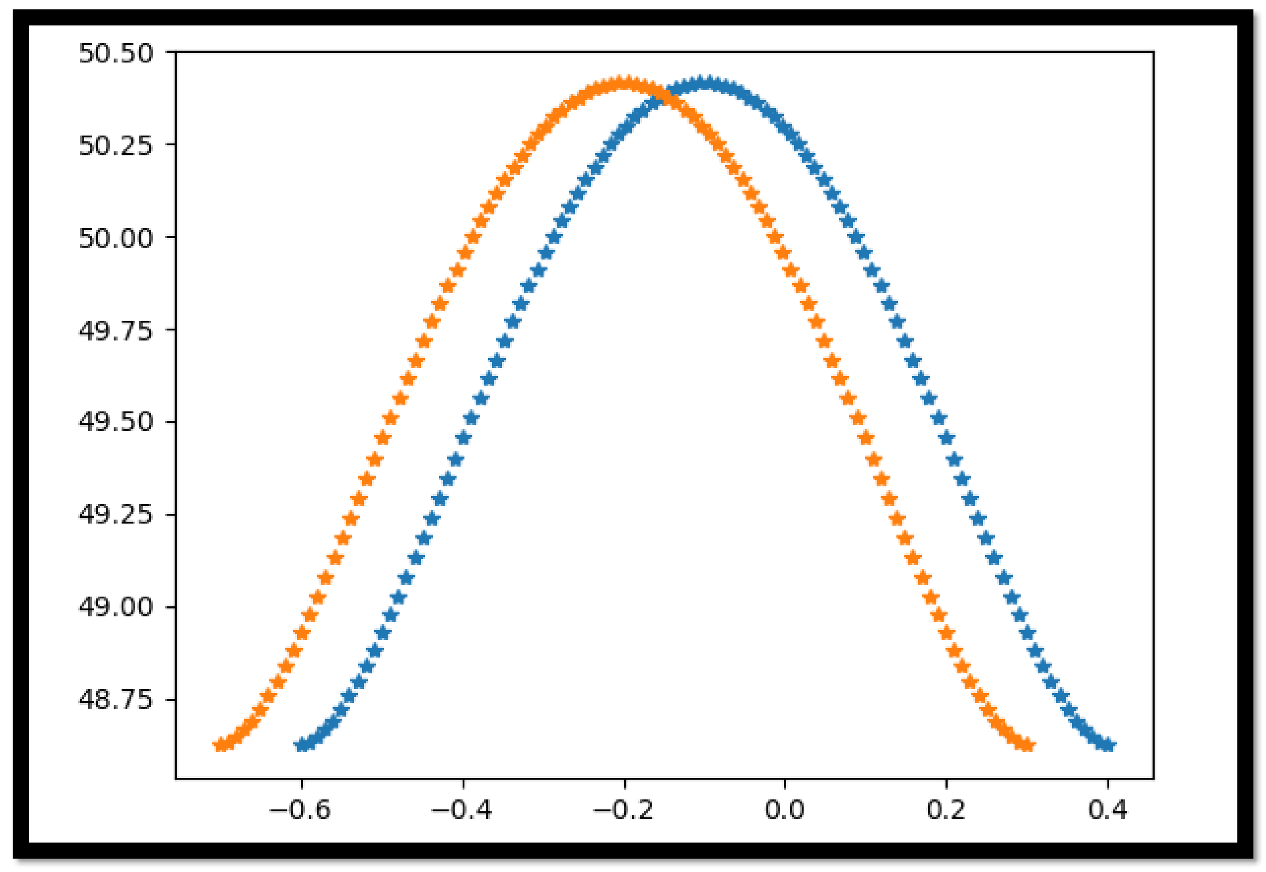

Fig. 2. Above shows a balanced, normal distribution if a user decides to keep migration in and out of the reach equal. Downstream biased migration moves the curve to the left. Upward-stream biased migration moves the curve right.

Results

Results are available in a variety of formats:

➢ A Histogram Plot.

➢ Textbox showing statistical data.

➢ Table showing raw data for each trial run.

➢ CSV, PNG files if user desires to export results

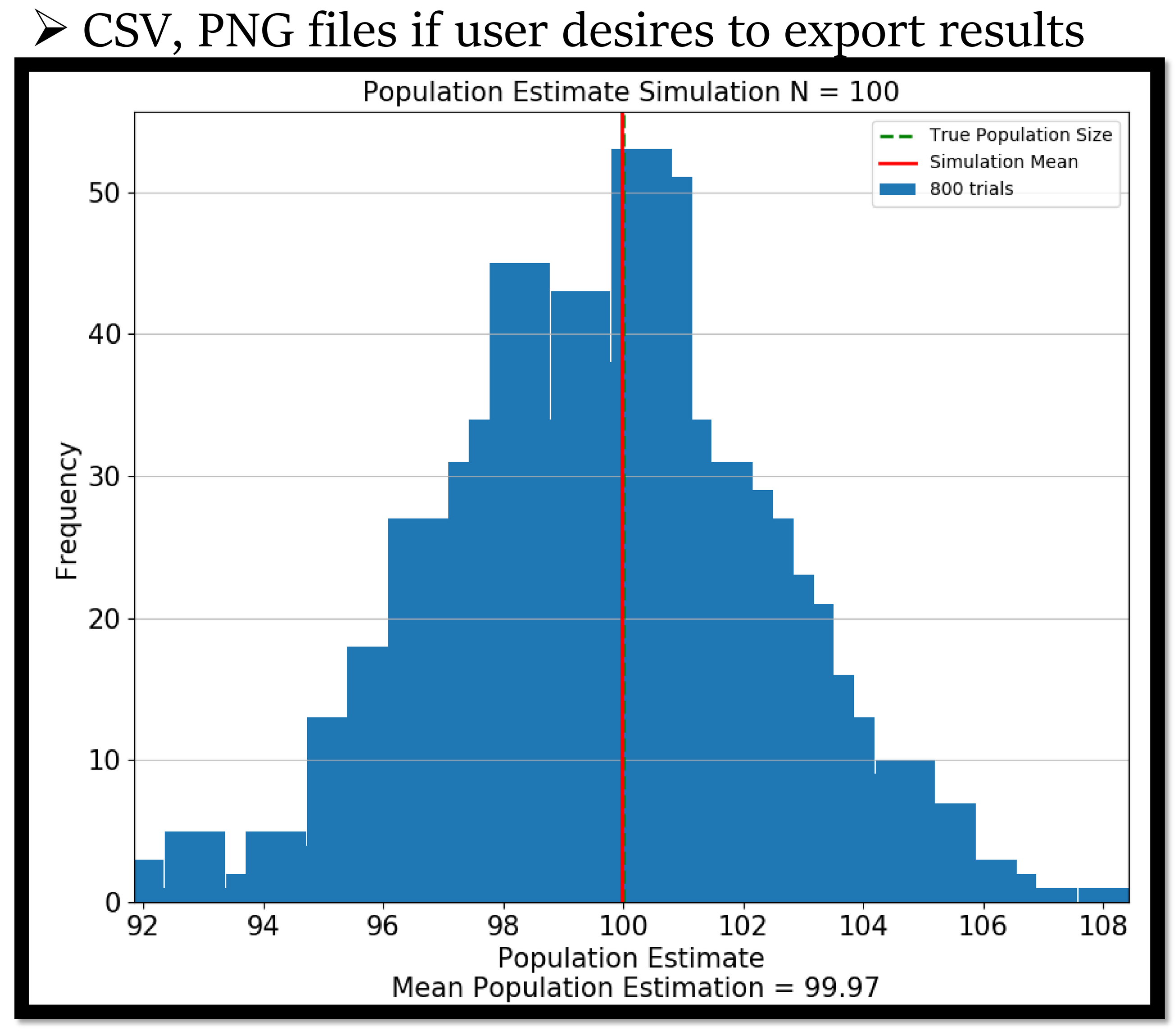

Fig. 3. A histogram plot to display simulation results.

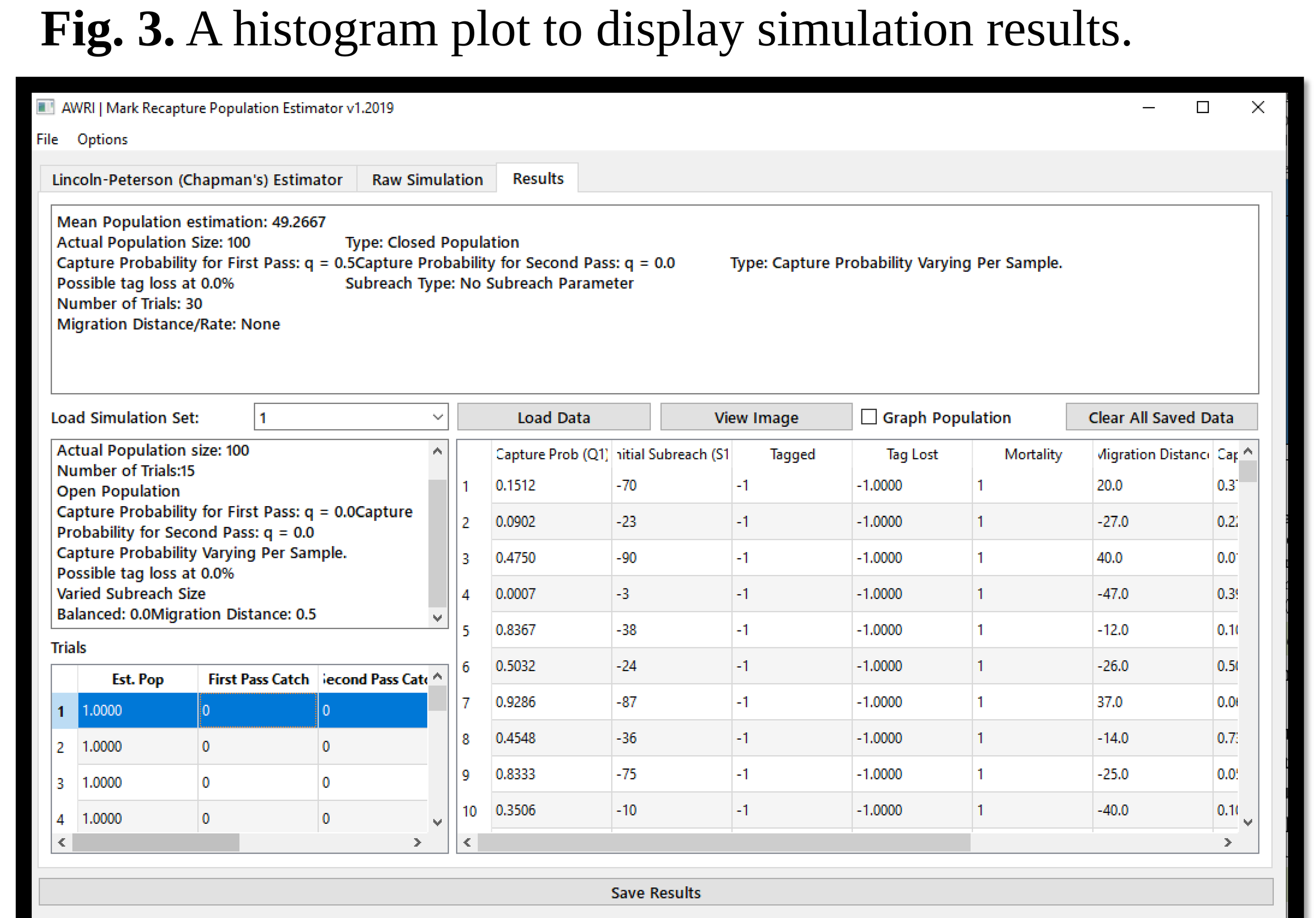

Fig. 4. The text box on the upper screen summarizes conditions of the most recent simulation as well providing statistical analysis.

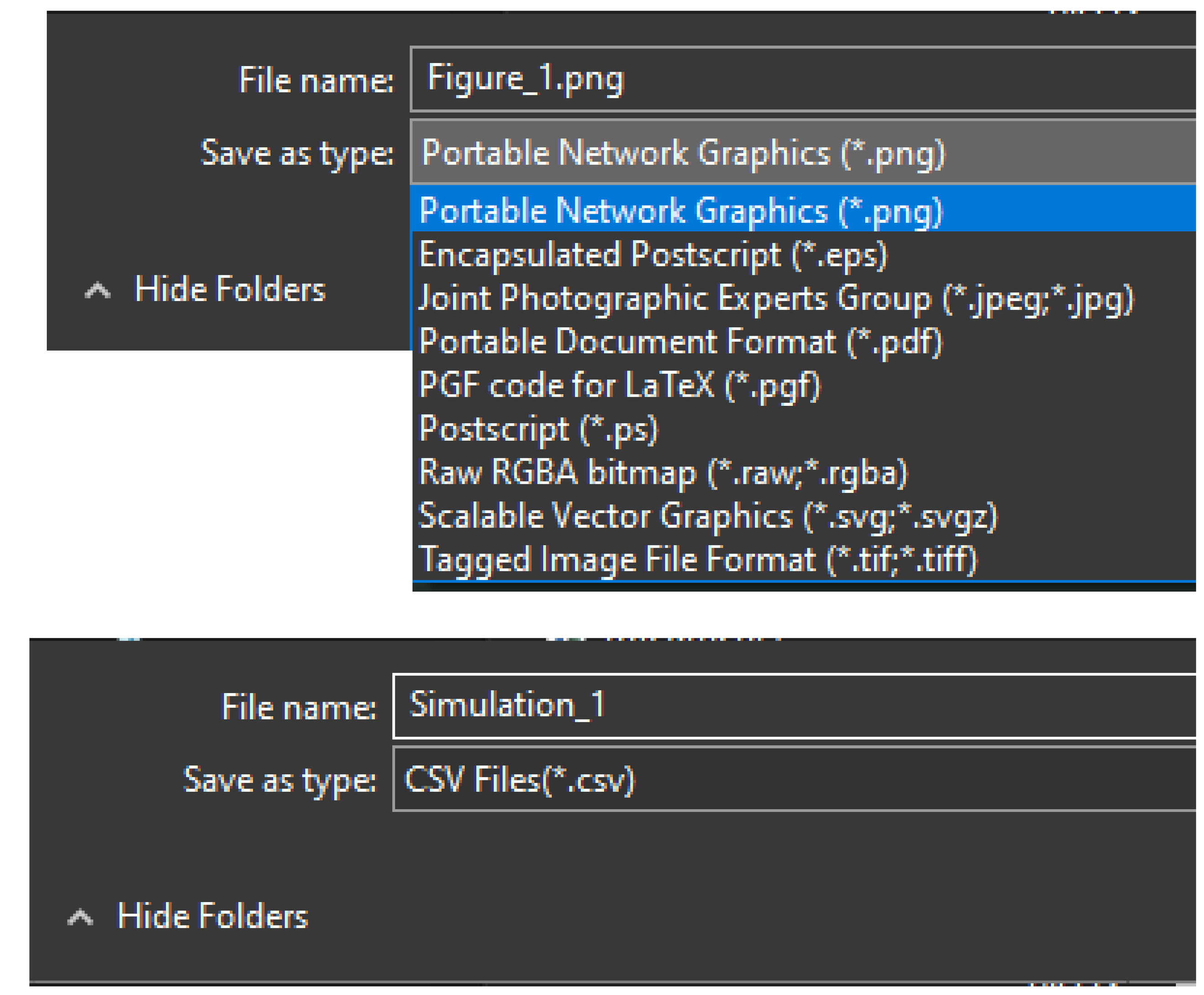

Fig. 5. Filetypes available to users to save their simulation results.

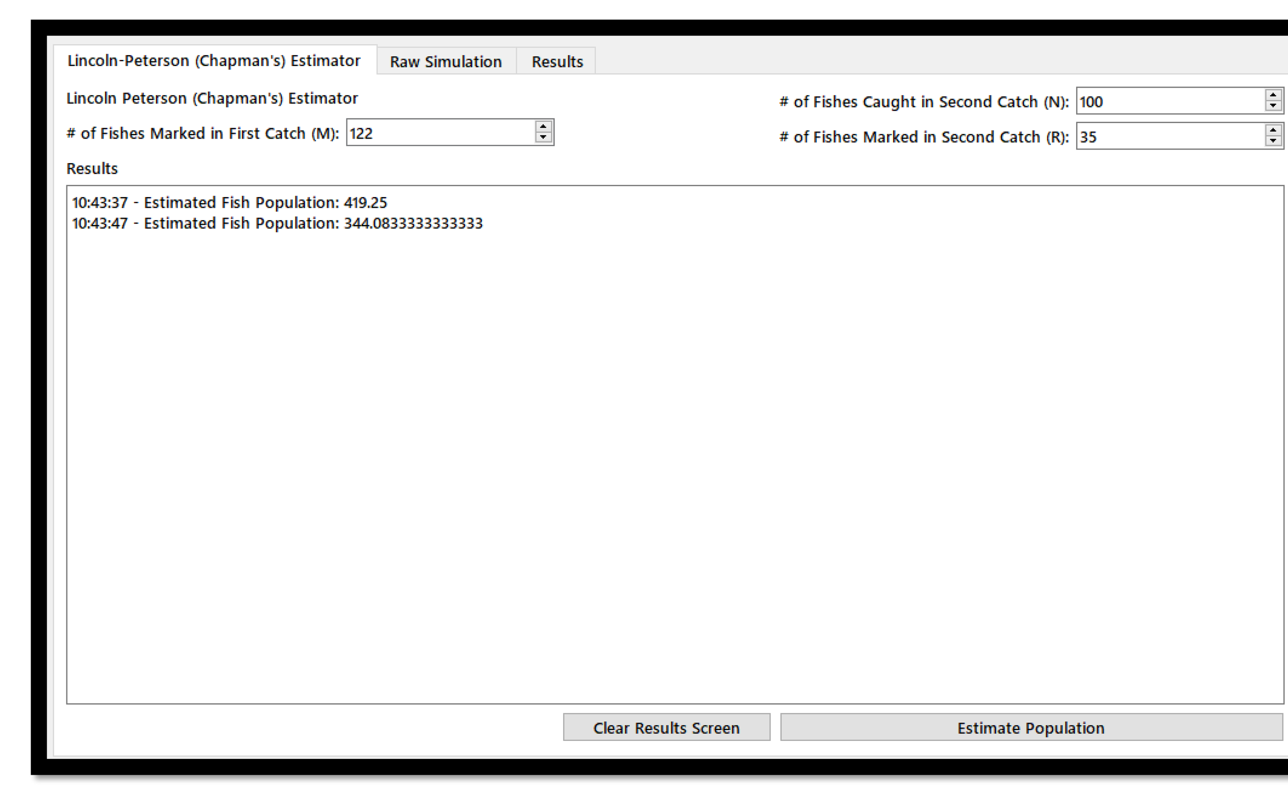

Fig. 6 Additional Tab allowing generic mark-recapture estimation calculations.

Technical Specifications

- Executable for Windows Platforms.
- Open-sourced code available on GitHub.
- Multiprocessing to speed up simulation times.

Conclusions & Future Improvements

- Add visual simulation feature.
- Refactor code to add in data structures and algorithm to further speed up simulation speed.

Acknowledgments

This research was made possible by the Bill & Diana Wipperfurth Student Research Scholarship. Special thanks to Grand Valley State University Office of Undergraduate Research and Scholarship (OURS) for providing travel funding through the Academic Conference Fund.

References

- Hunter, J. D. 2007. "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95.
- McNair, J.N., C.R. Ruetz III, A. Carlson, and J. Suh. 2018. Reducing effects of dispersal on the bias of 2-sample mark-recapture estimators of stream fish abundance. PLoS ONE 13(8):e0200733.