TrawlExpert: Tool for Watershed Biological Research

Lab section: L01
Christopher W. Schankula
Student ID: 400026650
schankuc@mcmaster.ca

January 26, 2018

1 Objective

Statistical and visual tool for the analysis of water ecosystems, based on scientific water trawl data. Provides researchers with tools to analyze large datasets to find patterns in fish populations, including the plotting of historical population data on a map, analyzing population trends over time and finding subpopulations of a certain species, genus, family, etc.

2 Motivation

The diminishing of fish populations in the Great Lakes became a problem in the latter half of the 20th century, with the total prey fish biomass declining in Lakes Superior, Michigan, Huron and Ontario between 1978 and 2015 (**?**). Annual bottom trawl surveys involve using specialized equipment to sweep an area and are used to determine the relative temporal variation in stock size, mortality and birth rates of different fish species (**?**). These surveys are performed annually and often have hundreds of thousands of records, making manual analysis infeasible.

TrawlExpert will give researchers tools to filter through these large amounts of data by allowing them to search through data based on class, order, genus, family or species. This will help support scientific researchers and fishing companies as they study fish populations in order to launch initiatives to preserve fish populations and perform their business in an environmentally friendly way going forward. As more data is collected on an annual basis, TrawlExpert can easily be injected with the new data and will adjust and scale accordingly, combining the new data with the old data for continued analysis.

TrawlExpert will also analyze the trawl data to find connected subpopulations within the data, giving researchers tools to analyze the portions of the water body that contain different populations and even track these specific subpopulations over time.

The focus of the project will be to develop these unique data searching and querying tools as a first step in a complete trawl survey analysis. For a complete analysis, tools like stratified statistical analysis would be required by the researcher (**?**). For purposes of controlling the scope of this project, the implementation of advanced trawl survey scientific and statistical analysis tools will be relegated to future developments.

3 Prior Work

4 Input / Output and Proposed Solutions

4.1 Datasets

The test dataset that will be used for purposes of this project is the *USGS Great Lakes Science Center Research Vessel Catch Information System Trawl* published by the United States Geological Survey. Compiled on yearly operations taking place from early spring to late fall from 1958 until 2016, the dataset contains

over 283,000 trawl occurrences in the five Great Lakes, including the latitude and longitude co-ordinates and biological classification such as family, genus and species. A complementary dataset contains 45,000 entries detailing the operations and parameters such as equipment.

4.2 Outputs

This section describes in detail the types of data analysis and queries that TrawlExpert will be able to perform. Section **??** describes a potential use case for a researcher studying a body of water.

4.2.1 Output 1: Historical Distribution

Researchers will be able to query for historical population data of a certain genus, family, etc, outputted in text format or generated as a figure.

4.2.2 Output 2: Geographical Distribution

For a given species over a given timeframe, the program will output the locations of all records matching that query in terms of the recorded location at which the fish were found.

4.2.3 Output 3: Geographical Subgroupings

The third main type of output will be to classify a given search into subgroupings of highly clustered populations.

4.3 Family Subgroupings: A Use Case Example

The dataset presented in section **??** will be used as the main input in order to generate these outputs. The location and temporal data given in the dataset will be used to help generate outputs 1 and 2. In order to illustrate this, the following example use case will be presented and analyzed:

The researcher is studying the decline of all species in the family *Cyprinidae*. She has a large dataset of trawl data and wishes to obtain information about the related subpopulations in the data. Therefore, she uses the TrawlExpert to generate an output of type 3 which will give her the output she needs for her research. By recursing down a pre-built tree structure built from the data (described in more detail in section **??**), the program will determine which genera, species and potentially subspecies are included in this family of organisms. It will locate all entries for this expanded search. These entries each contain the latitude and longitude at which the sightings were made. The program will then cluster geographically close results and return a list of these subgroupings, either in text format or visualized as a figure.

The researcher can now continue her scientific analysis of the data having intelligently and easily narrowed down to relevant data.

5 Algorithmic Challenges

As this project is data-driven and should be able to handle the very large datasets needed for scientific analysis, it presents many algorithmic challenges which can be broken into three categories: searching, sorting and graph algorithms.

5.1 Searching Algorithms

A modified form of binary search will be used for quickly locating specific records in the large dataset, and will be a crucial building block of all three main types of output. The modification comes from the fact that it will likely be most useful to locate the first element of a given type in a sorted dataset. For example, if a researcher is wishing to find all the entries for the species *Perca flavescens*, finding the first record will then allow the program to find all the elements by either performing a linear walk over all the records after that one or by using another modified version of binary search which finds the rightmost element of a certain key.

For speed of lookups without have to search each time, a sorted version of the dataset may be cached for use in subsequent searches.

5.2 Sorting Algorithms

Sorting will be crucial for both ordering historical data in chronological order as well as being the basis for binary search to work, since it requires data to be sorted, for example, alphabetically by species. The Mergesort algorithm will be advantageous due to its fast and predictable runtime ($N \lg N$) and its ability to handle sequences with many entries of a given key, of which we have many for our purposes.

5.3 Graph Algorithms

Graph algorithms will be important to the advanced searching features of the program. The first use will be in the analysis of groupings of organisms. The kingdom, phylum, class, order, family, genus, species and subspecies classification of each organism in the dataset allows us to build a detailed tree from which species in the same genus, for example, can be located. Secondly, a graph algorithm will be used to find connected components for generating output of type 3 described in section **??**. Points will be connected together based on their distance to surrounding points, and a connected components algorithm will be used to determine these groupings.

6 Project Plan

The following milestones will help inform our progress towards completing the goals. The team should be divided up into two subteams for maximum efficiency:

Milestone	**Subteam A**	**Subteam B**
Milestone 1 (End of Week 1)	Finished parsing module for .csv data to create Java objects that can be used for analysis; start data cleansing	General binary search module underway
Milestone 2 (End of Week 2)	Cleansed the data to remove or correct entries not containing all of the columns; start generating biological classification tree module	Finished and tested binary search; start mergesort
Milestone 3 (End of Week 4)	Finished and tested classification tree; start data visualization or formatted text output tools	Finished and tested mergesort; start writing query module for output 1, using mergesort and binary search to get results from data
Milestone 4 (End of Week 6)	Continue data visualization tools	Continue query module: finished output 1, start working on output class 2
Milestone 5 (End of Week 8)	Finished data visualization tools; start using them to display output from output 1 and 2	Finished output class 2

6.1 End of Week 1: Data Parsing

By the end of week 1, one small team should be able to parse our data into a Java-usable data structure. Another small team should start writing the general sorting and searching modules as well.

6.2 End of Week 2: Data Cleaning

Some entries in the data are not perfectly formatted in the correct columns. By the end of week 2, the data parsing team should have finished coming up with a plan to deal with this data, whether that means attempting to correct it or simply disregarding these datapoints.