2019

Multi-Modal Interfaces for Sensemaking of Graph-Connected Datasets

Anthony Wehrer
University of Central Florida

MULTI-MODAL INTERFACES
FOR
SENSEMAKING OF GRAPH-CONNECTED DATASETS

by

ANTHONY WEHRER
B.A., Rollins College, 2011
M.S., University of Central Florida, 2014

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2019

Major Professor: Charles E. Hughes

ABSTRACT

The visualization of hypothesized evolutionary processes is often shown through phylogenetic trees. Given evolutionary data presented in one of several widely accepted formats, software exists to render these data into a tree diagram. However, software packages commonly in use by biologists today often do not provide means to dynamically adjust and customize these diagrams for studying new hypothetical relationships, and for illustration and publication purposes. Even where these options are available, there can be a lack of intuitiveness and ease-of-use. The goal of our research is, thus, to investigate more natural and effective means of sensemaking of the data with different user input modalities. To this end, we experimented with different input modalities, designing and running a series of prototype studies, ultimately focusing our attention on pen-and-touch. Through several iterations of feedback and revision provided with the help of biology experts and students, we developed a pen-and-touch phylogenetic tree browsing and editing application called PhyloPen. This application expands on the capabilities of existing software with visualization techniques such as overview+detail, linked data views, and new interaction and manipulation techniques using pen-and-touch. To determine its impact on phylogenetic tree sensemaking, we conducted a within-subject comparative summative study against the most comparable and commonly used state-of-the-art mouse-based software system, Mesquite. Conducted with biology majors at the University of Central Florida, each used both software systems on a set number of exercise tasks of the same type. Determining effectiveness by several dependent measures, the results show PhyloPen was significantly better in terms of usefulness, satisfaction, ease-of-learning, ease-of-use, and cognitive load and relatively the same in variation of completion time. These results support an interaction paradigm that is superior to classic mouse-based interaction, which could have the potential to be applied to other communities that employ graph-based representations of their problem domains.

I dedicate this work to my wonderful family and friends without whom this would not have been possible. This applies especially to my mother who supported me in every way possible, my two academic advisers without whom I would have been lost, my very best friends who showed emotional support throughout, and my late grandfather Thomas Tancott who was very proud of me and I wish could have lived to see this day.

ACKNOWLEGMENTS

I would like to acknowledge the early contributions of the Arbor Workflows team in making this research possible, particularly Luke Harmon from the University of Idaho, Chelsea Specht from Cornell University, Robert Thacker from Sony Brook University, Jorge Soberon from the University of Kansas, Wes Turner from Simquest, Inc., and Jeff Baumes from Kitware, Inc. I give additional thanks to Luke Harmon for his participation in early prototype evaluations with his graduate and undergraduate students and Chelsea Specht for her continued interest in and advice given toward the PhyloPen project. I would also like to thank Eric Hoffman, Linda Walters, and the rest of the biology department at the University of Central Florida without whom our later prototype and summative studies would not have been possible. I give a very special thanks to Curtis Lisle of the Arbor Workflows team and KnowledgeVis, Inc., and my advisor Charles Hughes who have helped me through this process every step of the way. Finally, this work was supported by the National Science Foundation (DEB1208912). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: INTRODUCTION

1.1 Motivation and Objective

A *phylogenetic tree*, or *phylogeny*, is a visual representation of a hypothesized evolutionary hierarchy of descent. With the leaves as the current set of species or genes, the parent nodes are the supposed ancestor species or gene from which the two children split from one another in evolution. Each interior node of the tree is thus a hypothesized ancestor from which all children in the subtree, or *clade*, have descended. There can sometimes be more than two children per parent if not enough information is known to infer an exact order of descent. A branch in the tree can sometimes also have a *branch length* assigned to it, a numerical measure of the degree of divergence. These diagrams are used to help derive meaning from observed biological data, sometimes revealing patterns that are not obvious otherwise. Such patterns can elicit or support hypotheses, leading to new discovery about how life has evolved on Earth. An example phylogeny of Darwin's finches from the Galapagos Islands (genus Geospiza) is shown in Figure 1. This diagram is called a *cladogram*, which is an evolutionary tree with branches of equal distance.

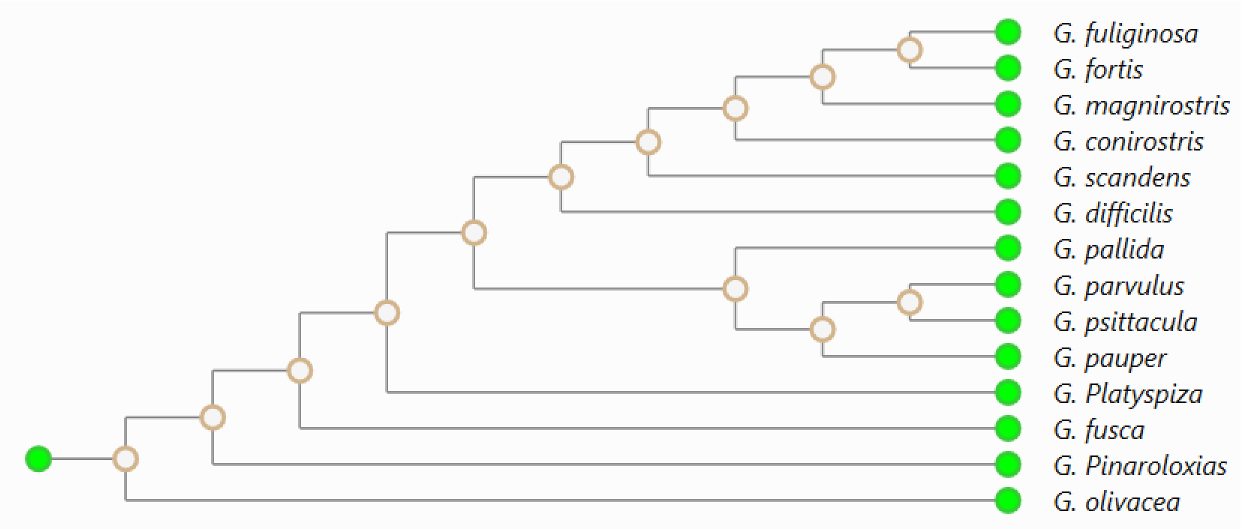


Figure 1: An example phylogenetic tree of the genus Geospiza birds rendered as a cladogram.

To help illustrate these sorts of patterns, biologists often require some means of rendering these visualizations. However, most existing phylogenetic tree rendering and analysis packages, such as Mesquite, FigTree, BEAST, and Mr. Bayes, provide renderings of tree diagrams that are static or limitedly interactable, and, where interactability exists, the means of interaction is not always straightforward. Due to these difficulties, when experts need a tree diagram for presentation and/or publication, they often settle for the time-consuming and arduous option of manually editing the diagrams with image editing software, such as Adobe Illustrator, most often requiring adjustments to branch colors and positions for proper illustration and species labels for readability. While lack of dynamic view manipulation limits what can be done, it can also limit the user's ability to make sense of the data. When dealing with large complex data visualizations, iterative manipulation of the view can help the user identify patterns more effectively and test their hypotheses by being able to retrieve the necessary details they need without being overwhelmed by all of it at once (Heer & Shneiderman, 2012). Therefore, it is our objective to provide biologists with an alternative interface for interacting with phylogenetic tree diagrams and study its effectiveness in sensemaking of the data compared to the field's current state-of-the-art in use.

A large part of this research is to investigate user interfaces manipulated by alternative input devices. Instead of traditional mouse and keyboard alone, we wish to study how alternative inputs such as pen and/or touch can affect the user's behavior, interest, and effectiveness at completing certain tasks. Analyzing the results, we can draw conclusions on the strengths and drawbacks of these alternative inputs and categorize where they help in completing the biologists' tasks in the most effective and natural way. The model of interaction we develop for phylogenetic trees could then be generalized to broader domains of use outside biology, especially to ones involving other forms of graphs.

For the dissertation, we started with some early experiments with alternative user interfaces for phylogenetic tree browsing, including the use of the Kinect, Leap Motion, pen, and touch. Using the results of the early experiments, we chose to focus on pen and touch together, specifically, and developed a more extensive tree browsing interface called PhyloPen (Wehrer, Yee, Lisle, & Hughes, 2015). This contribution is unique in that no other work has evaluated pen and touch together with the ability to annotate with them in a phylogenetic tree browsing program. After many iterations of prototype evaluations and improving the program through feedback, we developed an interface to be compared with the state of the art, the most popular and comparable being a modular system for evolutionary analysis called Mesquite. Our primary target audience in our user studies consists of biology students who have knowledge of phylogenetic trees but limited backgrounds in the manipulation of these structures. We evaluate how the two interfaces, PhyloPen and Mesquite, support the effectiveness of these subjects as regards sensemaking according to several measures, including completion time, usefulness, satisfaction, ease of learning, ease of use, cognitive load, and task completion.

1.2 Dissertation Overview

The proposal is split into several chapters: Chapter 2 presents related work. Chapter 3 covers the early experiments and the development of our first pen and touch user interface model. Chapter 4 discusses the current pen and touch interface modified to address the issues uncovered by past users. It also includes a summary of current contributions and goals to accomplish for the dissertation. Chapter 5 provides details regarding the research hypotheses we tested, the design of the experiments we ran, the data we collected, and the analyses we carried out along with how and when these results support our hypotheses. Chapter 6 concludes with a summary of results and a discussion on how our work can be used in broader domains.

CHAPTER TWO: RELATED WORK

2.1 Current Tree and Hierarchy Rendering Tools

At present, any members of the phylogenetic biology research community use R-based packages for analysis and rendering of phylogenetic trees and associated character matrices. The seminal package in this space was APE (Analysis of Phylogenetics and Evolution in R) (Paradis, 2006). APE includes functions to both analyze evolutionary data and perform phylogenetic tree rendering. Figure 2 shows the result of rendering the Geospiza dataset displayed earlier with APE.

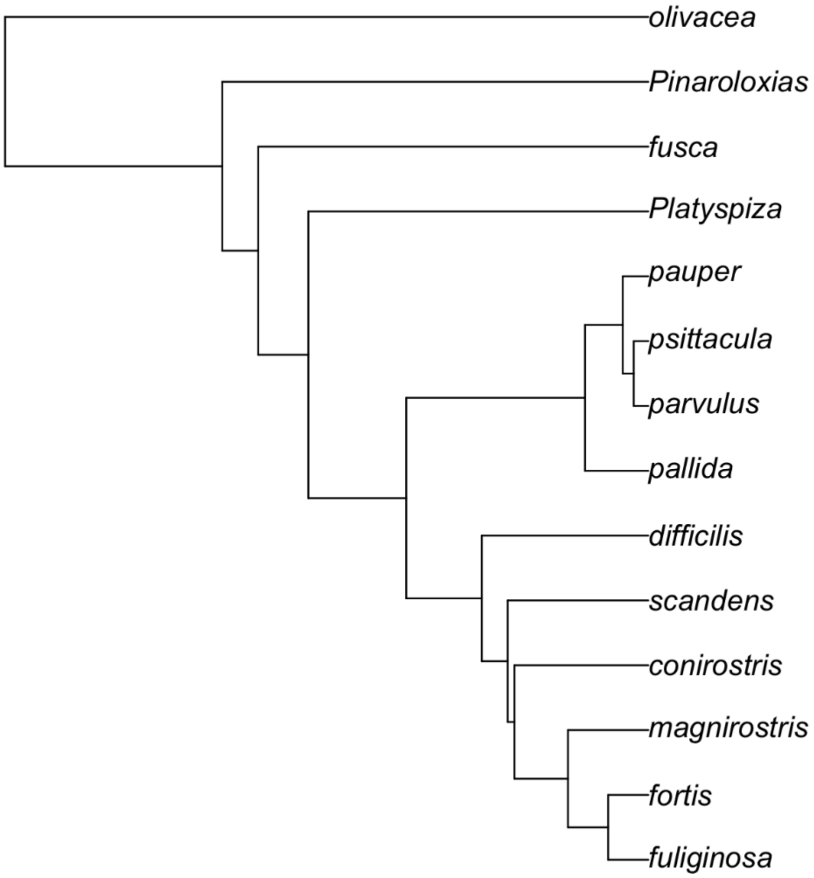


Figure 2: The genus Geospiza (Darwin's finches) rendered by APE, a non-interactive but popular plotting tool used by phylogenetic biologists.

These plots are easy for biologists to quickly generate using command-line plotting commands in the R language, but the rendering technology does not scale well to large trees or provide a means for the user to explore only portions of a large dataset. To illustrate this, the Anolis lizard tree (with 100 tips or leaf nodes) and the Choroflexi bacteria phylum (containing 12377 tips) are shown in Figure 3. It can be observed that, for larger datasets, the plotting overstrikes both letters and tree edges and becomes unreadable, at least using the default rendering options.

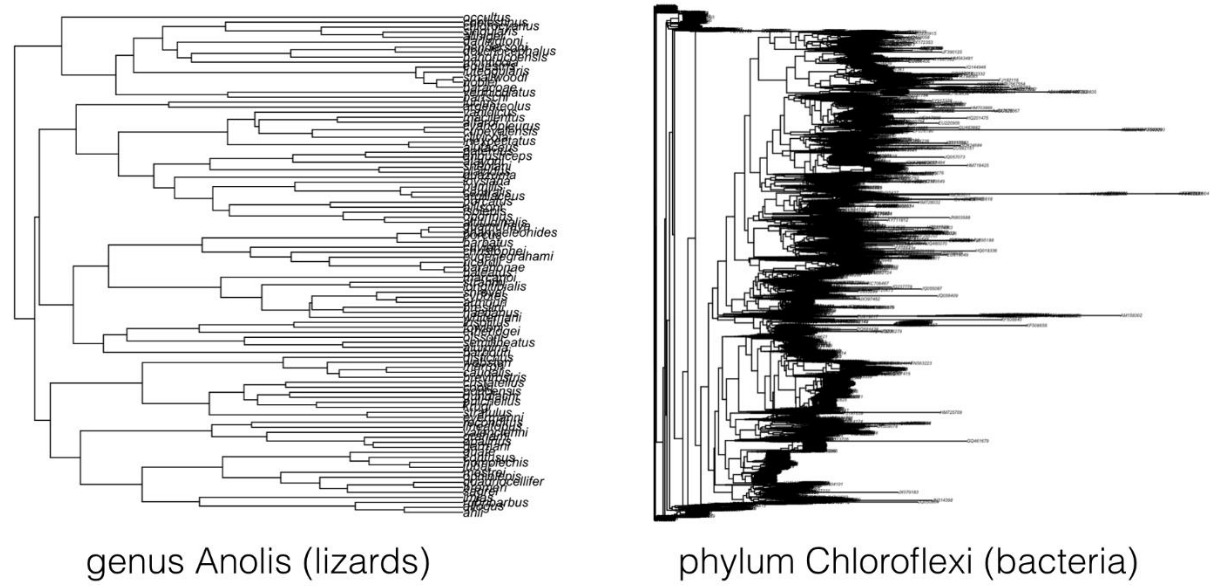
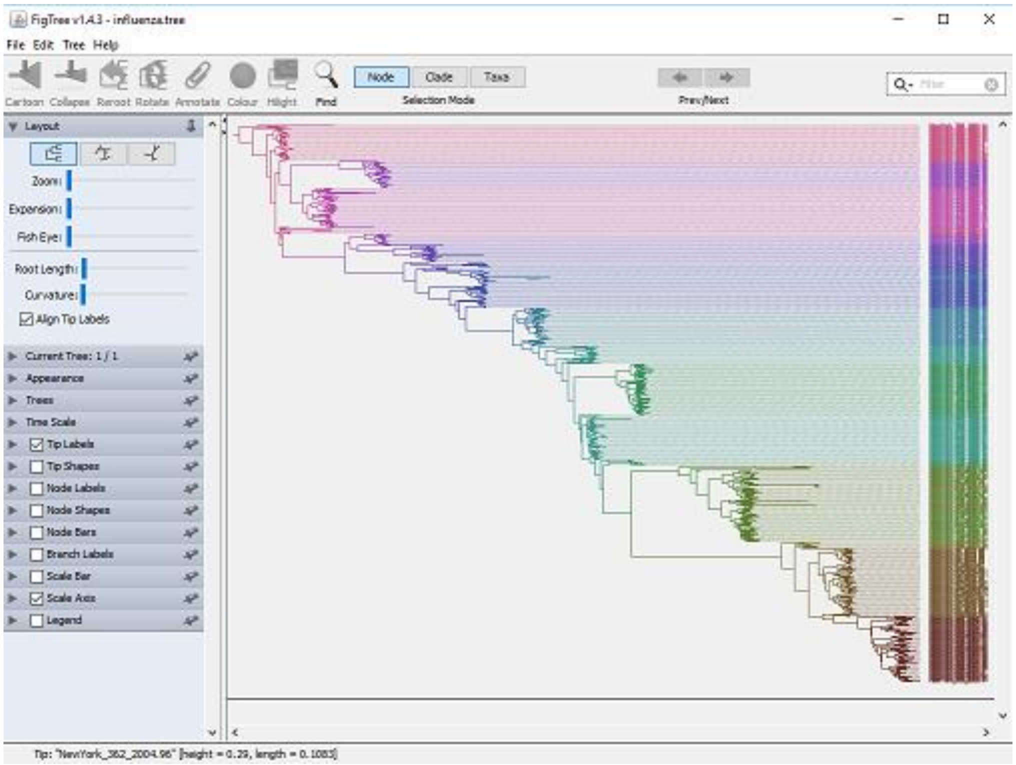


Figure 3: Plotting with APE for larger datasets. Details become obscured for large datasets.

The R language contains a designed-in plotting subsystem. Many datasets and datatypes inherit methods to *plot*() themselves. APE uses the plotting subsystem to render its trees by declaring plot methods that support phylogenetic trees. R also includes several plotting extension packages, one of which is called ggplot2. The ggplot2 function is of interest because it is used by other tree rendering packages and is based on the Grammar of Graphics (http://ggplot2.org/). The biology community is constantly developing new rendering technologies, but since they are based on this common plotting technology, these systems still generally offer limited or no interactivity.

One such R package that uses ggplot2 and allows extensive customization features is called ggtree (Yu, Smith, Zhu, Guan, & Lam, 2017). This package is designed specifically for the visualization and annotation of phylogenetic trees. The ggtree package provides commands to layout and display a tree and import many different formats of biological data (analyses) that can be associated with the tree as annotations. Additionally, the user can color parts of the tree by attribute values, create associated heatmaps of the attributes, highlight clades with colored rectangles, and annotate select clades in the tree with a vertical bar and label to the right of the tree spanning over its space. There are also commands for tree manipulation, such as collapsing and expanding clades, zooming to a portion of the tree, scaling clades, and rotating clades.

A more complete visualization tool used by biologists today with some similar features is FigTree (Rambaut, 2016). As opposed to being a package for visualization, FigTree is a full graphical user interface for viewing phylogenetic trees and is designed for producing publication-ready tree diagrams. The user interface allows for a fair amount of customization to the tree as illustrated in Figure 4. The user can change the appearance of nodes and branches and their labels, highlight clades in the tree, re-root the tree, annotate, choose whether to display certain labels, change the scaling of the tree, rotate clades, collapse and expand clades, and choose different layout styles altogether (rectangular, polar, and radial). Different graphical commands can be limited to certain parts of the tree by selecting relevant parts of the tree. The user can select the branch or label they want directly or drag the mouse to select an area with a rectangular box. The selection result will vary based on what mode the user has selected. It can either select per click nodes, whole clades, or taxa (any group of species or genes). One limitation of this viewer, however, is its navigation system. Regardless of what is selected, zooming will always zoom to the middle of the screen. The user can only zoom in to a specific point by moving a slider control on the side menu and scrolling over to the specific point of interest. This can be a tedious process when looking at large trees. Additionally, rendering large trees slows the program and causes it to lag a lot. Overall, FigTree does not scale well to large tree navigation.



Zoom:
Expansion:
Fish Eye:
Root Length:
Curvature:
☑ Align Tip Labels

Current Tree: 1 / 1
Appearance
Trees
Time Scale
☑ Tip Labels
□ Tip Shapes
□ Node Labels
□ Node Shapes
□ Node Bars
□ Branch Labels
□ Scale Bar
☑ Scale Axis
□ Legend

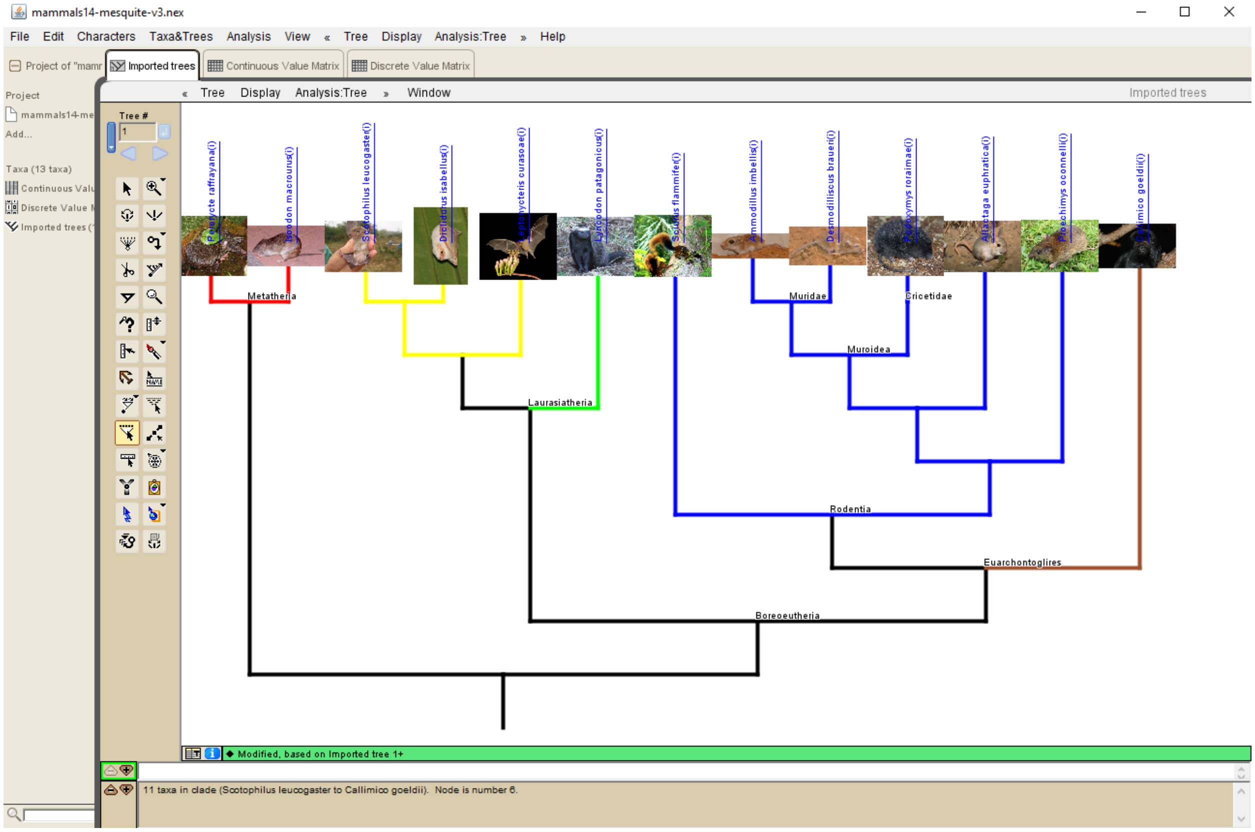
Tip: "NewYork_362_2004.96" [height = 0.29, length = 0.1083]

Figure 4: The FigTree phylogenetic tree viewer

While FigTree does not handle large datasets well, there is a newer Javascript-based visualization library that is made specifically for visualizing large datasets called D3.js. Although it has not historically been used extensively in biological visualization, it was used a fair amount in the initial development of the Arbor Workflows NSF project (Harmon, et al., 2013; Cranston, Harmon, O'Leary, & Lisle, 2014). Given a large dataset, it renders SVG graphics for visualization quickly and efficiently, which maintain a scale independence, allowing the graphics to maintain their quality at all scales while zooming. D3 can also avoid the issues with FigTree’s zoom slider with its built-in capability of using the mouse wheel to zoom to a point instead of zooming to the center. It can also handle touch events on objects in an HTML document instead of mouse. While it does not offer a phylogenetic-tree-oriented layout by default, it does have pre-written layout mechanisms for visualizing general hierarchical data. However, by changing the edge-link type, the resulting layout can be adjusted to resemble a cladogram.

While hierarchies are generally rendered with visible links between nodes, there can also be alternative layouts to consider such as one used by CiteVis (Stasko, et al., 2013), a simple user interface for the visualization of paper citations. Their concept is specifically narrowed to the IEEE InfoVis conference papers, laying out a series of circles in a row for each year to represent the papers from that year of the conference. Instead of connecting the nodes with links, their information about connections with other papers is represented with colors while interacting with the node. While hovering over a specific node, it turns yellow while papers in the conference that cite it become blue and papers in the conference that it cites become green. To keep this information visible even after removing the mouse hover, the user simply must click the node to select it.

Finally, the most widely used system and the one we will be focusing on for our study for comparison is a modular system for evolutionary analysis called Mesquite (Maddison & Maddison, 2019). Being modular, its capabilities for analysis can be extended based on the modules installed and are vast in potential. Among these different analysis methods is a means to render and edit a phylogenetic tree. An overview of the interface is illustrated in Figure 5. The illustrated tree was rearranged and customized with colored branches, pictures, and hyperlinks using Mesquite's modal command palette to the left of the tree. We summarize the command features most relevant to our goals in Table 1, including the ability to scale the tree or zoom into a specific clade, cut and reattach or delete a clade, annotate a clade, add nodes and branches, color branches, rotate clade child ordering, and perform web searches on species for more information about them.



Modified, based on Imported tree 1+

11 taxa in clade (Scotophilus leucogaster to Callimico goeldii). Node is number 6.

Figure 5: An overview of the Mesquite phylogenetic tree view. The character matrices can be found in the additional tabs.

With so many features, Mesquite is indeed a powerful tool for biologists, but it can also be complex and difficult to learn at times. Fortunately, tutorial files are provided to new users to understand the basics of how to use Mesquite, but even these tutorials have a limit to how much they teach about how to edit the phylogenetic tree renderings. It can take some experimenting to understand how to use the tool properly. Additionally, Mesquite suffers from a basic limitation in its tree rendering, specifically that it fits the tree size to the space available in the window. For larger trees, the images at the tips would occlude one another, and for even larger trees, it will become difficult to make out the details of the tree, a problem similar to the one illustrated in Figure 3. While the ability to zoom exists on the command palette, it may be difficult to initially find the point where the user wants to focus, especially when it is a fixed discrete increment zoom rather than one with continuous control.

Table 1: A summary of the most relevant command capabilities available in Mesquite.

Icon	Command & Description	Icon	Command & Description
🔍	**Zoom Image**: Click the tree to zoom in. Hold the Alt key while clicking to zoom out.	🔍	**Zoom into Clade**: Hover over and click a clade's incoming branch that you want to magnify and focus on. To stop focusing on the clade, click the clade's incoming branch again.
↖	**Cut and Reattach Clade**: Click and hold on the branch you want to move and drag the cursor to the branch you want to reattach it. The branch will be highlighted when you are successfully over it.	✂	**Prune Clade**: Hover over a branch and click it to prune it from the tree. To delete it from everywhere, go to its corresponding row in the character matrix, click its row number to select the row, drop down the Matrix menu, click Delete Selected Chars or Taxa, and confirm.
⊛	**Annotation**: You can annotate a region of the tree by putting the annotation on the branch leading to the clade. Just click the branch to add the annotation text. To show the annotation text, hover over it or display it by right clicking on the annotation icon for a context menu to click "See Notes On Tree."	⤢ NAME	**Add Node/Branch**: Click a branch to add a node to split it with. Additionally, select the second icon to then click the appropriate branch and name the node.
🖌	**Color Branches**: Click on the branch to color it. To change the color, right-click on the icon and go to "Branch paint color." You can also click "Remove color" to make the paint brush uncolor instead of color or remove all colors on the tree at once with "Remove all color from branches."	⟲	**Child Order Swap**: The branch interchange tool allows you to interchange two branches. To rotate a clade with it, you need to click and drag from the first branch you want to swap around and keep dragging until the illustrated line reaches the second branch you want in the swap. Release, and they should be swapped if done correctly.
↖	**Quick Web Info**: With this icon selected, you can click on the name of the species and be directed to a quick Wikipedia webpage about the species in a web browser.	↖	**Web Database Search**: For additional information about the species, you can try searching different databases about the species by clicking on the name while the icon is selected. You can change the search database by right clicking on the icon.

2.2 Interactive Hierarchical Displays

While most tree rendering systems in use by biology experts today generate static renderings of trees, there have been a few efforts to mitigate this limitation. OneZoom (Rosindell & Harmon, 2012) is one such effort. The interface uses identically shaped tree branches branching off from one another in a fractal pattern to represent the branching of species evolution in the tree. To be more specific, at the top level of the tree, one branch represents the base. Each child clade is represented with a smaller branch of the same shape that branches off from the parent branch. If, however, the child is a leaf, it is instead represented with a graphical tree leaf branching off. The fractal layout is rendered at each level in this manner. From far away, details about sections of the tree are hidden, but as the user zooms closer to it with the mouse scroll wheel, more information and detail is displayed about that section of the tree. The automatic detail changing avoids the labels and pictures about higher levels in the tree occluding the details that the user is trying to view farther down, displaying only what is relevant to the user's view. Since the display is not overloaded with information, this recursive fractal layout can handle large phylogenetic trees. OneZoom's fractal layout is illustrated in Figure 6. Also, although the focus is on traditional modes of input, the OneZoom interface has been used as a museum display on touch screens to explore the tree in the place of a mouse with a scroll wheel.

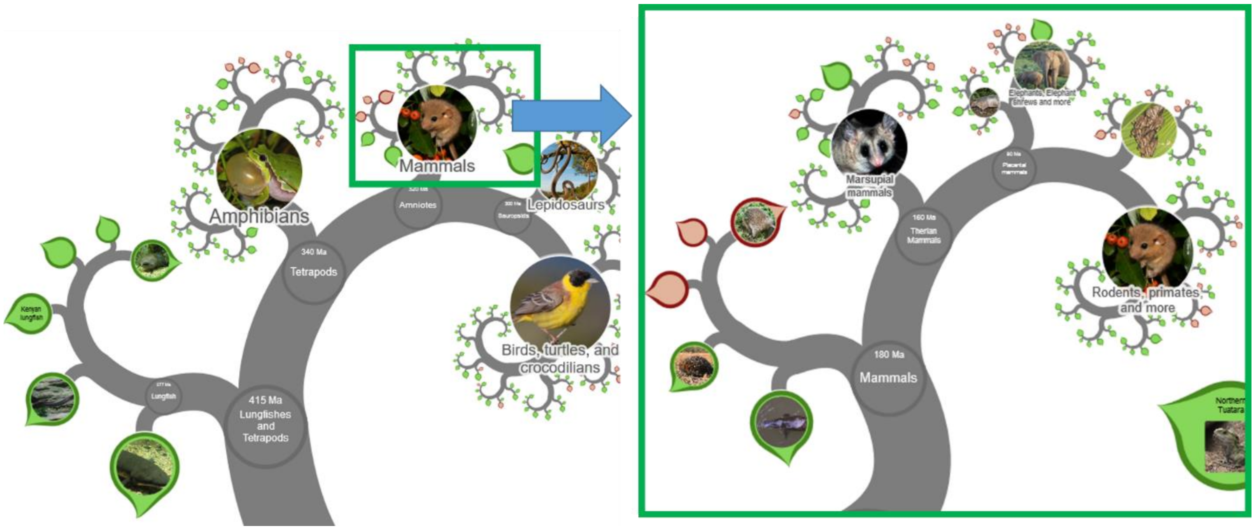


Figure 6: An illustration of the fractal layout used by OneZoom. When you zoom in on the area in the green rectangle, its information expands in the same shape.

Another interactive traditional-input-focused phylogenetic tree display tool that can handle large trees is the TreeJuxtaposer (Munzner, Guimbretiere, Tasiran, Zhang, & Zhou, 2003), a system specifically designed for comparing and browsing large trees. It is notable in the context of our own work in that it gives a novel means to view parts of a large tree without losing context with their “Accordion tree” navigation. This rectangular Focus+Context technique uses distortion of the layout by expanding and contracting rectangular areas of the tree both vertically and horizontally (much like an accordion expands and contracts). The surrounding areas shrink to ensure the visibility of the enlarged area of interest, allowing the user to see it in more detail. Since the areas that are shrunk are still visible, the user does not lose their context in exploring a large tree. This mechanism is illustrated in Figure 7.

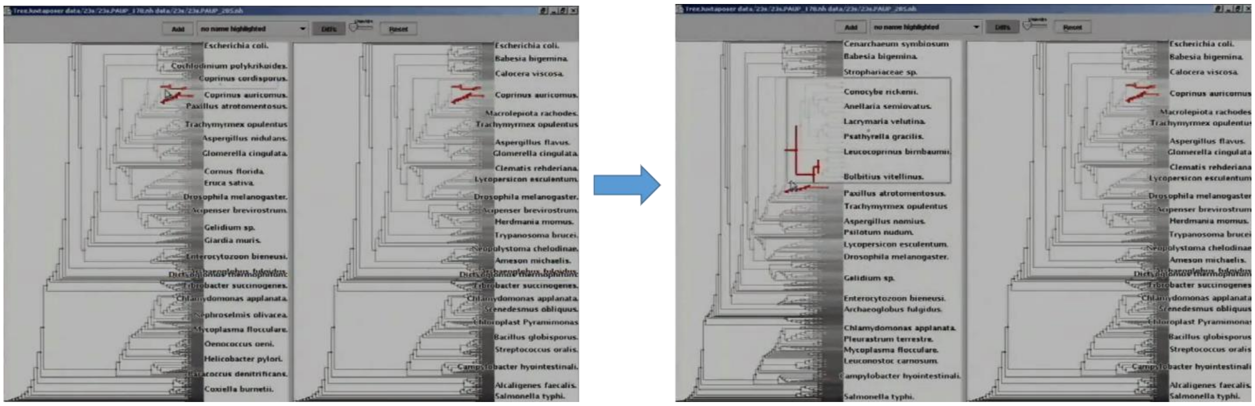


Figure 7: An illustration of the Accordion tree navigation in TreeJuxtaposer. By expanding a box on one area of the tree, the surrounding area is compressed so that the selected area of interest can be seen more clearly.

Aside from the traditional mouse and keyboard inputs, there have also been several interactive displays that take advantage of non-traditional means of input, many of which are outlined in a survey by Isenberg and Isenberg (Isenberg & Isenberg, Visualization on interactive surfaces: a research overview, 2013). Related to our own experiments, there have been a few touch-oriented interfaces, such as Involv (Horn, Tobiasz, & Shen, 2009) and the DeepTree exhibit (Block, et al., 2012) for interactive museum displays of species data and Phylo-Genie (Schneider, et al., 2012) for collaborative learning. There has also been an interface that integrates both pen and touch for co-located collaborative comparison tasks of different phylogenetic trees (Isenberg & Carpendale, 2007), specifically designed for phylogeny experts. While more of the work was focused on touch gestures, it presented a novel use of creating different types of annotations, whether by virtual sticky notes or by free-floating annotations integrated into the canvas, on a shared collaborative display. The study of the use of a stylus in phylogenetic tree browsing is rare or nonexistent otherwise.

2.3 Combining Input Modalities

While most interfaces in the domain focus on only one input modality (e.g., touch), our studies have focused on using different modalities together, specifically pen and touch (with some use of the keyboard). Previous studies demonstrate potential benefit in using pen and touch together. Specifically, in a study by Brandl et al. (2008), experiments were performed using touch, pen, and pen and touch combined. The results suggested that combining pen and touch in the gesture sets was superior in terms of speed and accuracy while also being preferred by most users. Another study that was performed by Walny et al. (2012) determined that there is a certain beneficial division of labor between pen and touch. While touch was found to be more natural for users in performing physical metaphor gestures, such as resizing and translating, pen was better used for writing, drawing, and gestures that do not have an intuitive physical correspondence in the real world. By combining the two inputs, it was also shown that explicit mode switching can be minimized, reducing the burden on the user and the amount of time it takes to complete tasks.

2.4 Ink Gesture Recognition

Ink gestures can be troublesome to recognize due to many factors. Given a set of points that form a polyline in two-dimensional space, we must interpret meaning from them to execute the user's desired action. However, writing with a human hand can be generally inconsistent and far from precise, especially when devices that have a stylus do not exactly mimic our normal writing surfaces. Some surfaces can feel like the user is gliding across an ice rink. This slip can create errors, especially when first putting the stylus down to write, creating extra unintended points at the endpoints of the ink polyline that are sometimes called "hooks." There can be other issues such as duplicate points and unsmooth lines. In interpreting the data, there are other complications to consider as well. The ink could be drawn at different scales, positions, and rotations (at an angle from the baseline). One gesture could contain multiple strokes, and each stroke could be drawn in different orders or directions. With so many factors and potential issues to consider, ink recognition can be a complex problem.

In response to the complexity of this problem, there have been many proposed approaches to ink gesture symbol recognition. Techniques have included support vector machines (Gorgevik, Cakmakov, & Radevski, 2001; Teow & Loe, 2002; Zhao, Liu, & Xia, 2000), Hidden Markov Models (Anderson, Bailey, & Skubic, 2004; Cao & Balakrishnan, 2005; Sezgin & Davis, 2005), neural networks (Pittman, 1991), linear classifiers (Rubine, 1991), template matching (Kara & Stahovich, 2004; Kristensson, 2004; Wobbrock, Wilson, & Li, 2007; Anthony & Wobbrock, 2010; Vatavu, Anthony, & Wobbrock, 2012), elastic matching with dynamic programming (Tappert, 1982; Myers & Rabiner, 1981), AdaBoost (Freund & Schapire, 1996), and feature-based heuristic methods (Notowidigdo & Miller, 2004). A more comprehensive, but somewhat out of date, survey of techniques is provided by Srihari & Plamondon (2000).

When we first started this work, our focus was more on feature-based heuristic methods. One of the most prominent features of a potential gesture or symbol to analyze is the number of corners between lines. One algorithm for finding corners is ShortStraw (Wolin, Eoff, & Hammond, 2008), commonly chosen for its accuracy and ease of implementation. An extension was made that improves the technique’s accuracy and handles some of its limitations called IStraw (Xiong & LaViola Jr., 2009). In addition, an add-on method extension is proposed in their work that can handle strokes that have curves or arcs. While using corners to identify gestures avoids the issue of having to consider the scale, rotation, and stroke orderings/directions, unwanted corners may be detected from drawing errors, especially hooks, which can have many sharp changes in direction of the polyline. An extra corner could lead a program to process a triangle as a square instead, making good filtering and dehooking methods important. Additionally, feature analysis can be used to narrow possibilities for a more sophisticated algorithm.

More recently, our focus has shifted to template matching. Given several training example templates, the algorithm must match the drawn symbol with the most similar template. The most relevant family of algorithms to our work is the $ recognizers: $1 (Wobbrock, Wilson, & Li, 2007), $N (Anthony & Wobbrock, 2010), and $P (Vatavu, Anthony, & Wobbrock, 2012), methods similar in intention to ShortStraw and IStraw in that they were created to be fairly accurate and easy to implement. Each of these are geometric-based template matching algorithms, using an instance-based nearest-neighbor classification system measured with a Euclidean scoring function. Each predecessor builds on the others, but each can be suited for different uses. The $1 recognizer is for unistrokes (one-stroke gestures) while $N and $P are for unistrokes and multistrokes (multiple-stroke gestures). The $N recognizer is more accurate than $1 and is additionally stroke direction-invariant. $P eliminates the limitations and combinatorics overhead of $N and is more accurate, but it cannot be rotation-invariant like $N.

2.5 Retaining Context in Interactions

Large phylogenetic trees cannot practically be viewed all at once in full detail. The area covered is too large. To provide the user with more detail, we can allow zooming in to a specific area. However, when zooming into such a large area, it is possible for the user to become disoriented and forget their current place in the tree relative to the whole. This requires them to zoom out and keep checking their current location. Aside from zooming, there are other visualization techniques that can help avoid this issue. If the user had an overview view of the information as well, they would no longer lose their place. This is what characterizes overview+detail techniques in which two views of the visualized dataset, an overview and a detailed view, are spatially separated and maintained simultaneously on the display. A good example of such an interface in which the user is viewing part of a map with a small overview of the entire map in the bottom corner is presented in (Cockburn, Karlson, & Bederson, 2009). In the context of phylogenetic trees, this would mean a view with the entire tree without detail and a detailed view displaying some subset of the tree. Shneiderman discusses the importance of providing the user with an overview even in his early discussions of designing user interfaces (Shneiderman & Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction, 2005). In fact, his Visual Information Seeking Mantra suggests “overview first, zoom and filter, then details on demand” (Shneiderman, 1996). Synteny Explorer’s tree interface was designed with this mantra in mind (Bryan, et al., 2017). In the Phylo View, the overview+detail approach is applied with an overview picture of the entire phylogenetic tree and a second view where the user can “zoom and filter” by navigating through a part of the tree and selecting species of interest. The user can then receive “detail on demand” on the selected species by switching to the Genome View, which displays a visualization of the chromosomes and synteny mappings.

In trying to maintain context, there is also the approach of focus+context, a technique that minimizes the space between the views of overview+detail and displays the focus within the contextual view. A fisheye view is a common approach where the focal area is magnified and displayed within its surrounding context (Cockburn, Karlson, & Bederson, 2009). TreeJuxtaposer (Munzner, Guimbretiere, Tasiran, Zhang, & Zhou, 2003) is one of the few phylogenetic tree interfaces that uses this technique. The area of interest is magnified, and the surrounding area is distorted to fit the focal point in to context. There are also other general tree interfaces that have focus+context such as DOI Tree (Card & Nation, 2002), SpaceTree (Plaisant, Grosjean, & Bederson, 2002), and Graham's system (Graham & Kennedy, 2001). While Graham's system is similar in purpose, it is not as scalable to large trees as TreeJuxtaposer.

2.6 User Studies and Evaluation

An important part in creating user interfaces is measuring their effectiveness. As stated by Munzner (2008), a common pitfall in results is the "But My Friends Liked It" situation where a new technique receiving positive informal evaluation user studies results from a few lab mates and friends outside the domain of use is proposed to be enough to demonstrate its merit and usefulness. Without a properly created user study, results can often be misleading or not be enough to compel its receiving audience into believing in the merit of the technique. Thus, the interface is not validated to solve the target domain problem. Therefore, it is important to be careful in designing effective user studies that address the needs of the intended community of use.

Although it is critical to create an effective user study, there are no clear-cut methods of creating one. However, it is possible to observe past user studies of similar intent to design our own. The primary purpose of a phylogenetic tree diagram is to provide a visualization of the evolutionary process that promotes analysis and understanding. Consequently, one of the important points of evaluation is whether our user interface effectively promotes learning when used as a teaching tool. Observing user studies across multiple disciplines, there are many ways to approach this problem. The goal of the Synteny Explorer (Bryan, et al., 2017), for example, is to teach the user about the genome evolution between species. As a means of validating that the system achieved this goal, 60 volunteers from the student body of the University of California (one third being from biology) were given a multiple-choice quiz that required the user to make use of the tool to answer the questions, demonstrating their understanding of the information illustrated in the user interface. However, our interest is more in the classroom setting. For instance, the U.S. Air Force Academy promotes active learning in classrooms by integrating interactive visualization tools into their computer science curriculum (Schweitzer & Brown, 2007). While no formal user studies exist, there is an overwhelming volume of informal evidence in course evaluations and the general behavior of the students that demonstrate the effectiveness of the additional visual aid in the learning process. There was, however, a formal user study performed with high school students to investigate how they develop their understanding of chemical representations when provided with a visualization tool called eChem (Wu, Krajcik, & Soloway, 2001). Data was collected from 71 students over six weeks, and they were each tested with a pre- and post-test. The results demonstrated significant improvement in their understanding of the material. Video of the students' performances were additionally captured to analyze how eChem helped them. The only downside to the study was that lack of a control group to compare against.

Simulations serve a similar role to educational interactive visualizations where educational information is evolved from a static to dynamic context. Thus, it is also appropriate to observe user studies for simulations with the goal of teaching the user some concept or experience. Connected Chemistry (Stieff & Wilensky, 2003) is one example of such a simulation with a goal of providing deeper understanding of chemistry concepts and processes in a classroom (and laboratory) setting. It was specifically designed for an introductory high school chemistry course. While no full evaluation was performed in a classroom setting, a preliminary 6-student study was performed to test whether Connected Chemistry provided enough benefit to students already knowledgeable in the field to warrant a larger study. Another example is a user study that evaluates the learning performance on complex physiological models from the Haemodynamics Simulator

(HAEMOSIM) (Holzinger, Kickmeier-Rust, Wassertheurer, & Hessinger, 2009). In this study, 92 undergraduate students were recruited from the Medical University of Gaz and split between three groups, one that was provided with a two-page lesson printout, another that was given access to HAEMOSIM with a shortlist of basic functions, and another that was given access to HAEMOSIM after viewing a short clip on how to use the simulator. Each group was given 6 minutes to learn, as recommended by surgeons of the teaching hospital as a realistic learning condition for their profession. They were administered pre- and post-tests to test their understanding before and after. Results showed that while the groups that were provided with textual information did not differ significantly, providing a visual demonstration of the simulator's operation beforehand resulted in a group with a significantly higher mean test score.

In this chapter, we reviewed related work on static and interactive hierarchical rendering tools for phylogenetic trees, past studies that show benefit in combining pen and touch, various ink gesture recognition techniques, and techniques for retaining context in interacting with large trees. In the next chapter, we discuss the preliminary research work that spawned from these past works to arrive at our final project.

CHAPTER THREE: PRELIMINARY WORK

Since previous renderings of phylogenetic trees were mostly static, our project goals focus around rendering and interacting with phylogenetic trees that can be dynamically changed using novel interaction paradigms. We started by creating and evaluating web-based phylogenetic applications with alternative external devices due to their value to a research team distributed across several universities funded under the Arbor Workflows NSF project. The webpages used the D3 (Data-Driven Documents) Javascript library to render a tree because of its usefulness and efficiency in rendering and visualizing large datasets with scale independent SVG graphics (a feature that is significant for zoom quality). While D3 has layouts for trees and graphs, we used custom rendering layouts to match the style of traditional phylogenetic trees as shown in Figure 5.

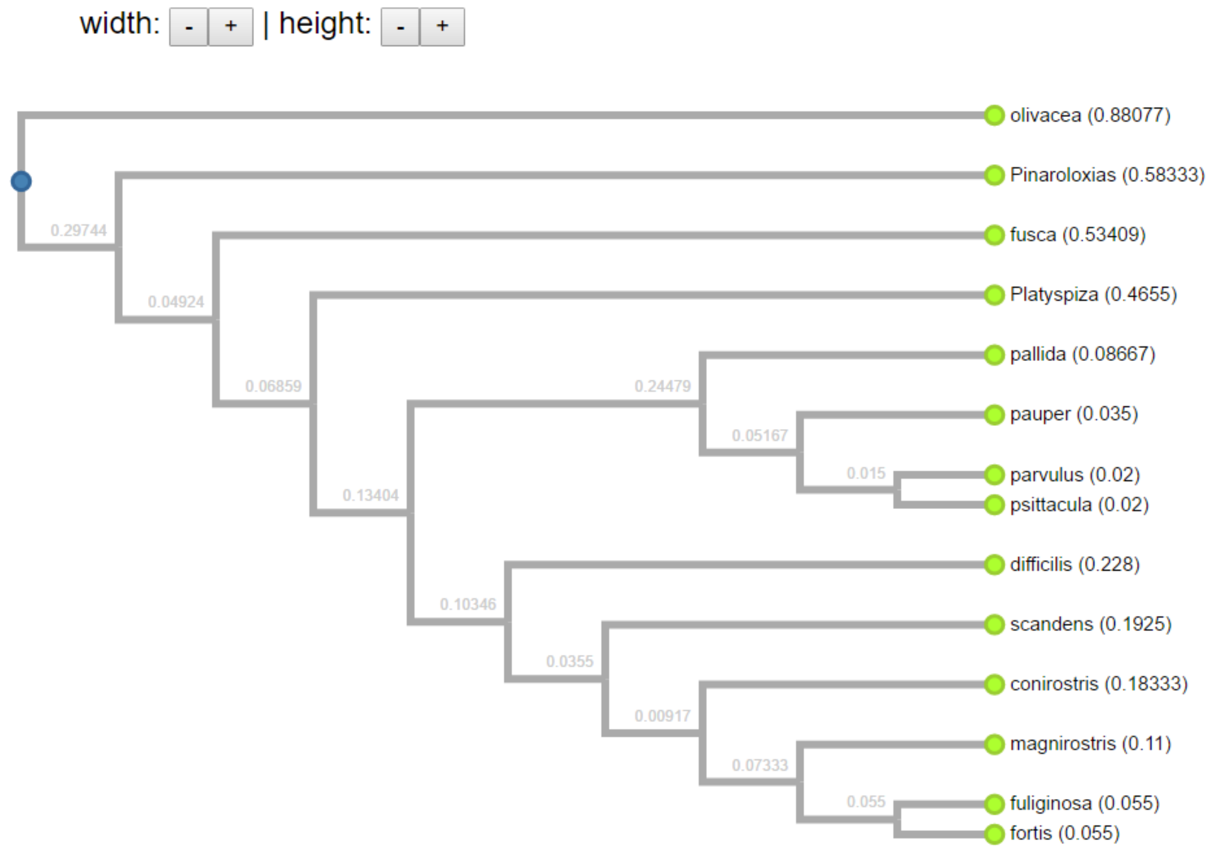


Figure 8: This cladogram is an early version of our work in rendering phylogenetic trees in D3 on webpages. Controls were provided to adjust the dimensions of the tree by some fixed amount, and the user could zoom in and out with the mouse scroll wheel and translate by dragging.

3.1 Hand Gesture Based User Interfaces

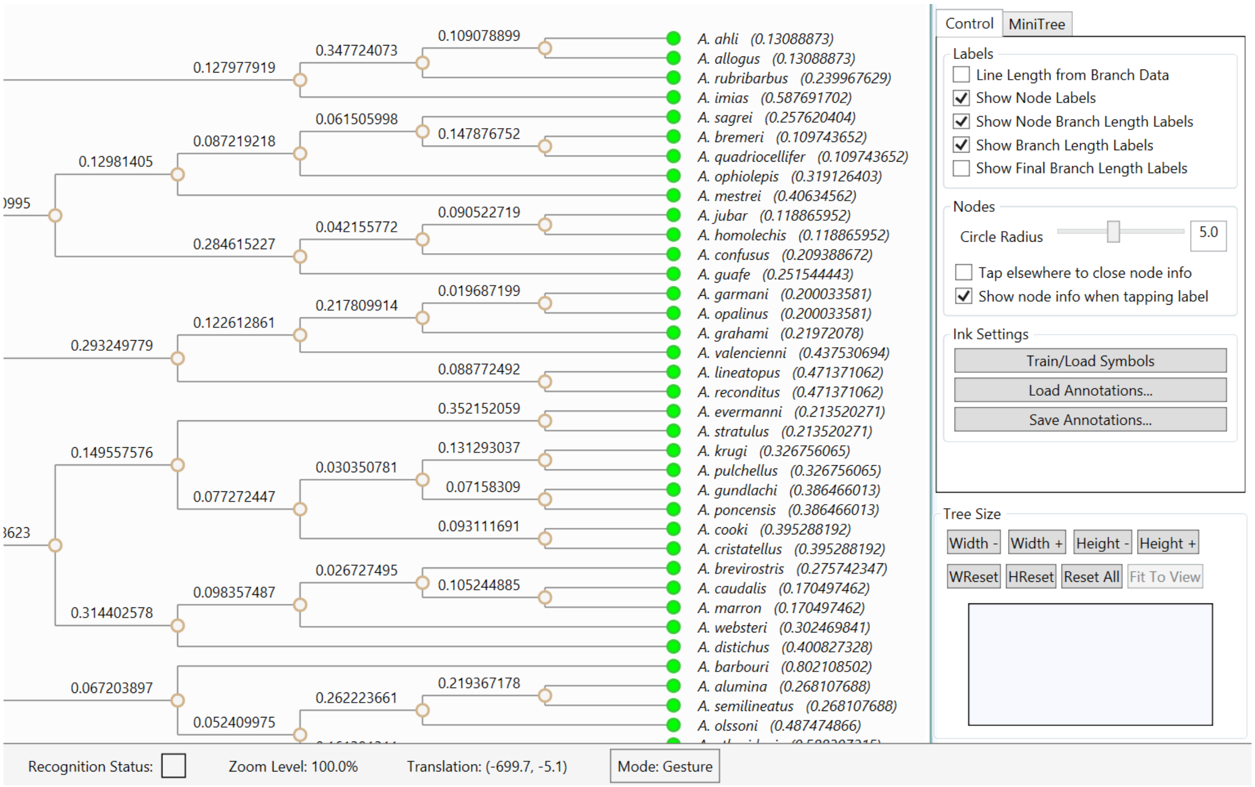
Our earliest experiments used the Kinect as the input device. As a web-based interface, commands deciphered from the user's movements had to be routed to the webpage with UDP packets from a local background program to a Tangelo web server (http://tangelo.readthedocs.io/). With Tangelo integrating Python services with the target webpage underneath, a Python web service could provide the webpage with the command data. The gestures were limited to hand tracking. Following the user's right hand in some arbitrarily decided rectangular region in front of the user, the program scaled this rectangular region to the webpage's local coordinate space, providing a cursor that follows along with the hand's movement. The user could hover over parts of the tree to see a tooltip about the data using their hand. To translate through the tree, the user made swiping motions in the direction they wanted to move, and to zoom to a part of the tree, the user held out their hands, facing one another, and moved them toward one another or away from one another to scale the tree accordingly.

There were various detrimental issues associated with this approach. One of the major issues was that the user's hand could not stay steady enough while being held out in front of them to hover over small details in the tree. The tooltip could not be maintained long enough for them to read the information. Another problem was tracking issues, especially when trying to perform the zoom gesture. If the user's hands were too close together, the Kinect would immediately lose tracking. With these issues, the user found it more frustrating and fatiguing to use. The other problem with the Kinect approach was the limits of gesture potential. While translating, zooming, and hovering made some intuitive sense, performing operations on the tree would involve more convoluted gestures that would be harder to recognize and/or harder to remember for the user due to lack of intuitiveness. This work led us to consider such devices as the Leap Motion with its greater detection accuracy and fine-precision hand gesturing. However, we ultimately opted for pen and touch user interfaces for reasons detailed in the next section. For the time being, we leave this investigation to future work.

3.2 Pen-Centric User Interface

We alternatively experimented with two other forms of input, namely pen and touch. This choice was inspired by a pen and touch user interface class taught by Dr. LaViola, starting out as a final project in the class. The initial pen and touch software was developed in C# with the Windows Presentation Foundation (WPF) due to its well-articulated built-in support for pen and touch events. While we were constrained to WPF as part of the class and we found that touch gesture events could be adapted to a webpage style using D3, we would later find that support for pen interaction was very limited in general for the web and even in commonly available programming packages. WPF also uses vector graphics just as D3, making it very useful for scaling quality. Therefore, using WPF in C# for our new development was an ideal solution at the time.

The layout of the graphical user interface consisted of three main sections: The viewport on the upper-left side that displayed the rendered phylogenetic tree and acted as a drawing canvas; a resizable sidebar on upper-right, which contained various features related to the phylogenetic tree rendering; and a status bar on the bottom to display the state of the user interface such as the zoom percentage, the translation point, and gesture recognition results. This user interface is illustrated in Figure 9. Upon starting the application, the user was provided with a list of phylogenetic trees that they could load into the interface from a MongoDB NoSQL database. New trees could be loaded into the database outside the program from standard input file formats such as Newick and phyloXML only by employing Python scripts.



Labels

☐ Line Length from Branch Data
☑ Show Node Labels
☑ Show Node Branch Length Labels
☑ Show Branch Length Labels
☐ Show Final Branch Length Labels

Nodes

Circle Radius 5.0

☐ Tap elsewhere to close node info
☑ Show node info when tapping label

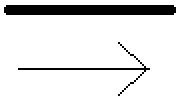
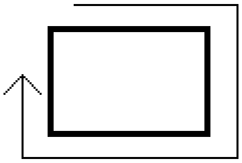
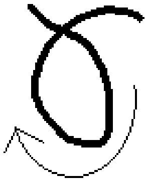
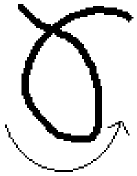
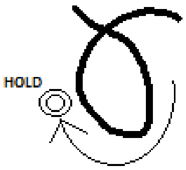
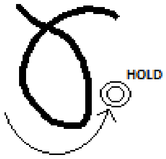
Ink Settings

Recognition Status: ☐ Zoom Level: 100.0% Translation: (-699.7, -5.1) Mode: Gesture

Figure 9: The standard layout of PhyloPen with the ink canvas phylogenetic tree view on the upper-left, the feature sidebar on the upper-right, and the status bar on the bottom.

Since our efforts were focused on pen gestures, our initial intent was to study the pen modality by itself and limit the program to only pen gestures, leading us to dub the interface as PhyloPen. There were several ink gestures for the first implementation, the most significant of which were dedicated to navigation. To translate the phylogenetic tree around the view, the user would draw a line whose length determined how far to translate. Once the gesture was registered, the view transitioned to the new location. Zooming, on the other hand, could be carried out in a few different ways. To zoom in and out, the user could create a loop in a clockwise or counterclockwise motion, respectively. The screen would then gradually zoom in or out by some fixed amount. The user could also hold at the end of the stroke to perform a continuous zoom and then release to stop. One additional feature for zooming was for the user to be able to define a new viewport on the diagram and have the view transition to that new viewport. This could be carried out by drawing a rectangle around the desired area in the tree. The implemented gestures are listed in Table 2.

Table 2: The initial set of ink gestures.

Gesture	Name & Description	Gesture	Name & Description
	**Translation**: Draw a line in the direction that the tree should be translated.		**Zoom to Region**: Draw a rectangle around a region to gradually translate and zoom to that region as the new viewport.
	**Zoom Out**: Draw a loop in the clockwise direction to zoom out a fixed amount.		**Zoom In**: Draw a loop in the counterclockwise direction to zoom in a fixed amount.
 	**Continuous Zoom Out**: Draw a loop in the clockwise direction and hold the stylus or finger at the end. Release to stop.	 	**Continuous Zoom In**: Draw a loop in the clockwise direction and hold the stylus or finger at the end. Release to stop.

In addition to buttons to change the tree size by a fixed amount in the bottom right corner of the interface, there was a small ink canvas on the sidebar to change the width and height of the tree structure, called the tree resize canvas. A horizontal stroke would change the width of the tree, increasing or decreasing, depending on whether the user strokes from left to right or right to left. Similarly, a vertical stroke would change the height of the tree. A line at any other angle would change both the width and height at the same time based directly on the distance traveled horizontally and vertically, as illustrated in Figure 10.

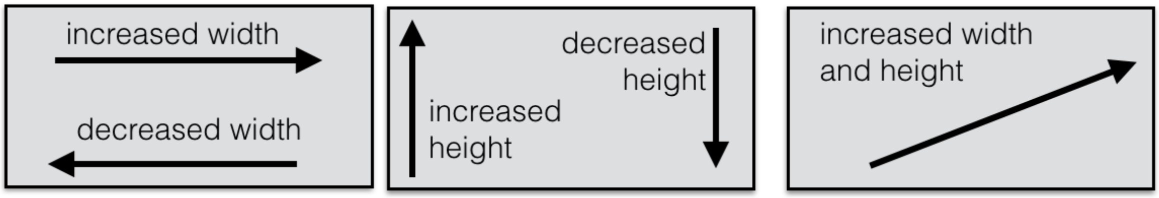


Figure 10: The tree resize ink canvas can change the width and height of the tree.

There were a few other important features, both directly and indirectly related to the use of the stylus. The user could hover over the leaf nodes in the tree to preview information about the various species and tap the nodes to keep the information displayed. Displayed information could contain characteristic measurements and data as well as a picture representation of the species. Another important feature was the miniature tree on the sidebar. It displayed a scaled static tree and helped the user know where their view position was relative to the overall tree with a small rectangle to represent the viewport. This rectangle changed in width and height while zooming in and out. Additionally, the user could move the viewport rectangle in the miniature tree to navigate to a new position in the tree. A PhyloPen browsing example for an influenza virus dataset using the miniature tree view is shown in Figure 11.

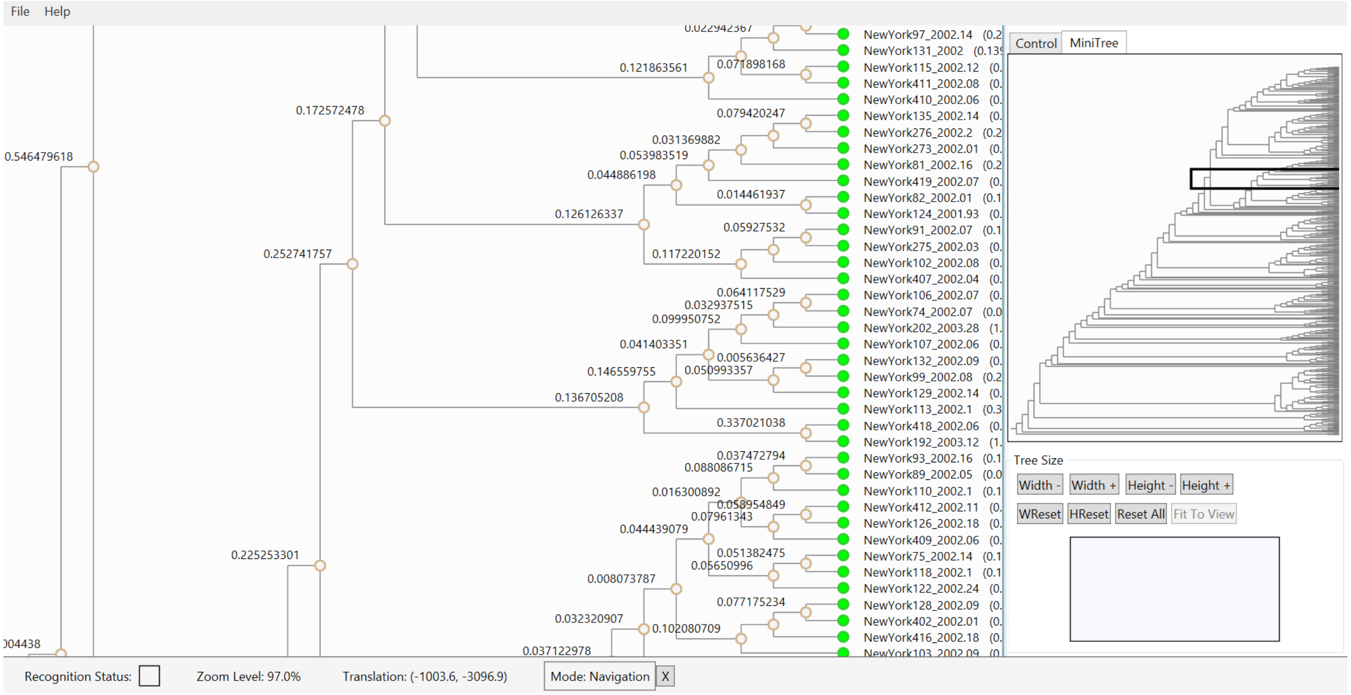


Figure 11: The miniature tree view at the top right provides location context.

For our first attempt at annotating nodes, the user could only add annotations to the leaf nodes. When the user opened the information box for a species by tapping a leaf node, a button appeared to add an annotation. Another box then appeared to the right of the tree where the user could write the new annotation. The box started out relatively small, but if part of the ink touched the inside of the box, it would expand to fit. When finished, the user would tap the appropriate button and the ink would be associated with the node. At the time, there could only be one.

3.3 Pen and Touch User Interface

After feedback from two leading evolutionary biologists who were most knowledgeable about the potential uses for a pen-based interface for phylogenetic trees and some personal testing with the features, several changes were made, the most significant of which being to introduce touch input into the interface.

The first major revision was trading most of the navigation ink gestures with touch gestures, specifically replacing the loop and line ink gestures with pinch to zoom and finger swipes for translation. Since a finger can be used to draw on the canvas by default, we needed a way to differentiate when the user was drawing and when the user was navigating. So, to begin navigation mode, we had the user place two fingers on the screen instead of one. The program would automatically exit navigation mode once both fingers were removed. Initially, if the user wanted to translate, they would remove one of their fingers and keep the other finger on the screen while swiping. However, it became cumbersome to have to put down two fingers each time the user wanted to swipe. Thus, we revised the protocol to stay in navigation mode until the user double tapped the screen. Also, an inertial component was included at the end of each swipe. Later, we found some users had the tendency to want to put the pen to the screen to perform gestures while they were still in navigation mode. So additionally, when the pen is touched to the screen, navigation mode is automatically exited. This one change turned out to save a lot of time in mode switching, allowing for a simple and instantaneous transition back to their previous mode.

The change from primarily ink gestures to primarily touch gestures for navigation was made for a few different reasons. First, the biologists felt that touch gestures are natural for most users, especially with the popularity of mobile devices with pinch and swipe touch gestures, and the benefits of multimodal pen and touch have already been established (Brandl, Forlines, Wigdor, Haller, & Shen, 2008) (Walny, Lee, Johns, Riche, & Carpendale, 2012). Second, having a line for translation limited what other multiple stroke ink gestures could possibly be implemented without having to make the ink gesture recognition explicitly modal. For example, if we were to use an "X" gesture to start annotation, the first line of the “X” would be registered as a translation without any means of differentiation. Third, there is an inherent delay in ink gesture navigation from having to first recognize the ink gesture to executing it. Touch gestures, however, take effect in real time. Fourth, when in annotation mode, the user was unable to use navigation ink gestures. This can be problematic if the user wants to look at something that is off-screen after already starting the annotation or if the user runs out of room while writing and wants to make more by shifting the view over.

The other major revision was in the process by which the user created annotations. The original auto-expanding box-subwindow method limited the user on where they could write annotations, and the box could not expand unless at least part of the stroke was inside the designated writing box. Instead, to begin annotation mode this time, the user drew an “X” gesture anywhere on the canvas. The user could then write the annotation where desired and double-tap the node with which they wanted to associate the annotation, a method we felt would be far more natural for the user. Once the annotation is associated with a node, a small colored annotation square then appears next to the node, displaying the annotation whenever hovered or tapped. This process can be repeated for any given node, and another annotation square will appear offset from any previous annotation square. It will also appear in a different color. Hypothetically, this should work well, assuming there are only a few annotations per node. When tapping on a set of multiple annotations, an annotation selection tool will appear to pick between them, as shown in Figure 12.

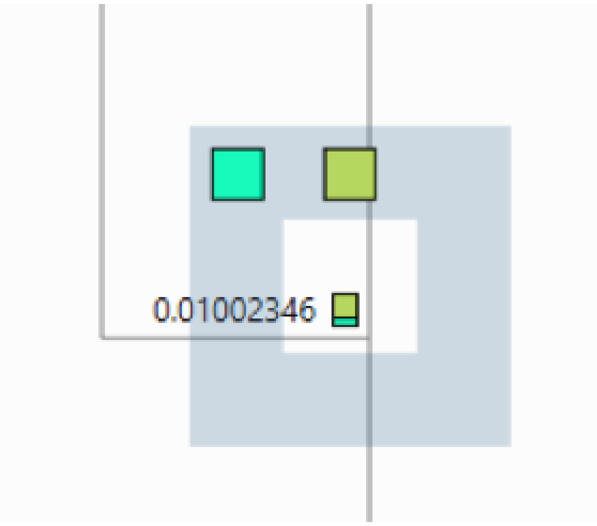


Figure 12: The Annotation Selector allows the user to select (by touch) the annotation to observe

Additionally, another new component was added to the annotation feature based on feedback from the biologists. It is not always the case that the user will want to associate the annotation with only one node in the tree. It could apply to a whole clade of species. For example, suppose at one of the branching points in a tree, the subsequent species all have tails. So, the user begins by writing “has tail” at the starting point and associating it with that node. Then, to associate it with the whole clade, or sub-tree, the user makes a swiping gesture on the node with either a finger or the stylus in a rightward motion (toward the leaves). Swiping in a leftward motion associates an annotation with all the node’s ancestors. To show when the annotation is associated with multiple nodes, the branches along the path will become highlighted when the annotation is previewed.

The other important features that originated with feedback from the biologists were the ability to swap a clade’s immediate children and to expand and collapse parts of the phylogenetic tree. We implemented the swap by drawing a “C” ink gesture at the point where a branch splits. The swap is illustrated in Figure 13.

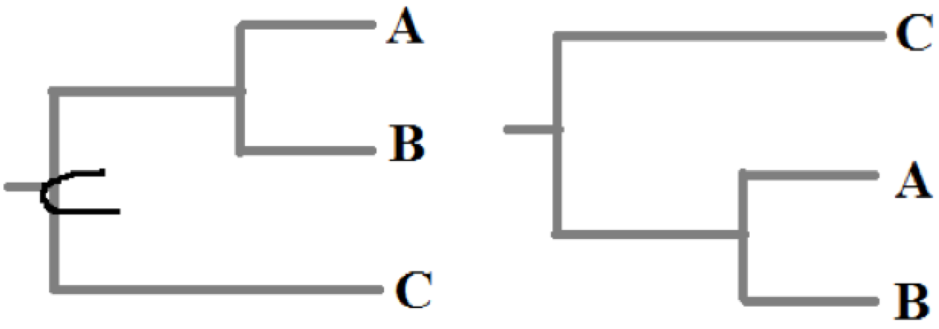
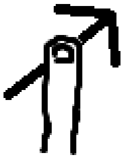
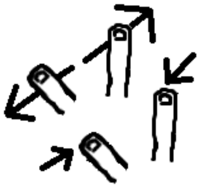
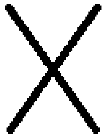
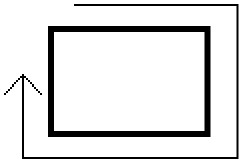
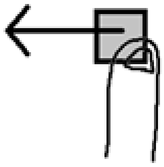
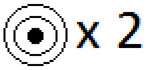
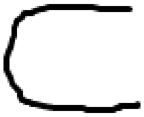
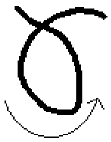


Figure 13: Pivoting or rotating children using a gesture around the clade.

The immediate children of the clade are swapped (AB and C becomes C and AB), but the lower children stay the same (AB does not become BA). This feature of the user interface is important for educational purposes to illustrate that the ordering is not significant. The expand and collapse operations were performed with the same gesture as was previously used to zoom with the pen. The user drew the loop gesture around the node in the tree they wanted collapsed or expanded. All these updated gestures are summarized in Table 3.

Table 3: The revised gesture set for PhyloPen

Gesture	Name & Description	Gesture	Name & Description
	**Translation**: Once in navigation mode after putting two fingers on the screen, swipe the screen with a finger.		**Zoom**: Once in navigation mode after putting two fingers on the screen, pinch two fingers to zoom in and stretch them apart to zoom out.
	**Annotation**: To begin annotating, draw an "X" in two crossing line strokes. Then, double tap a node to associate the annotation when done.		**Zoom to Region**: Draw a rectangle around a region to gradually translate and zoom to that region as the new viewport.
	**Annotation Flow to Ancestors/Descendants:** Copy annotation to all parent/child nodes of an annotated clade by making a swiping motion over the annotation rectangle to the left/right using a pen or finger.		**Display Node Details:** Double tap a node to display more information about a species.
	**Clade Child Swap**: Draw a "C" like symbol over the clade root to swap its immediate child ordering.		**Expand and Collapse Clades**: Draw a loop around the clade-node to be expanded or collapsed.

3.4 User Evaluation

A small prototype user study was conducted on PhyloPen to evaluate the interface model's usefulness and usability (Wehrer, Yee, Lisle, & Hughes, 2015). The participants consisted of 3 Ph.D. students and 3 faculty members from the University of Idaho, each of whom were given a copy of PhyloPen to independently test on a Microsoft Surface Pro 2. The participants were given an accompanying video that demonstrated how PhyloPen worked and were asked to respond to a post-questionnaire that listed an array of questions that would provide feedback on how to direct development on PhyloPen. The participants were additionally asked to rate each feature of PhyloPen on a 5-point Likert scale in terms of usefulness and usability. Usefulness ratings were labeled as “hardly useful,” “a little useful,” “pretty useful,” “really useful,” and “extremely useful.” Usability ratings were labeled similarly except with the word “fun” in the place of “usable.” The average usefulness and usability ratings are shown in Figure 14 where the black bars indicate the standard deviation range.

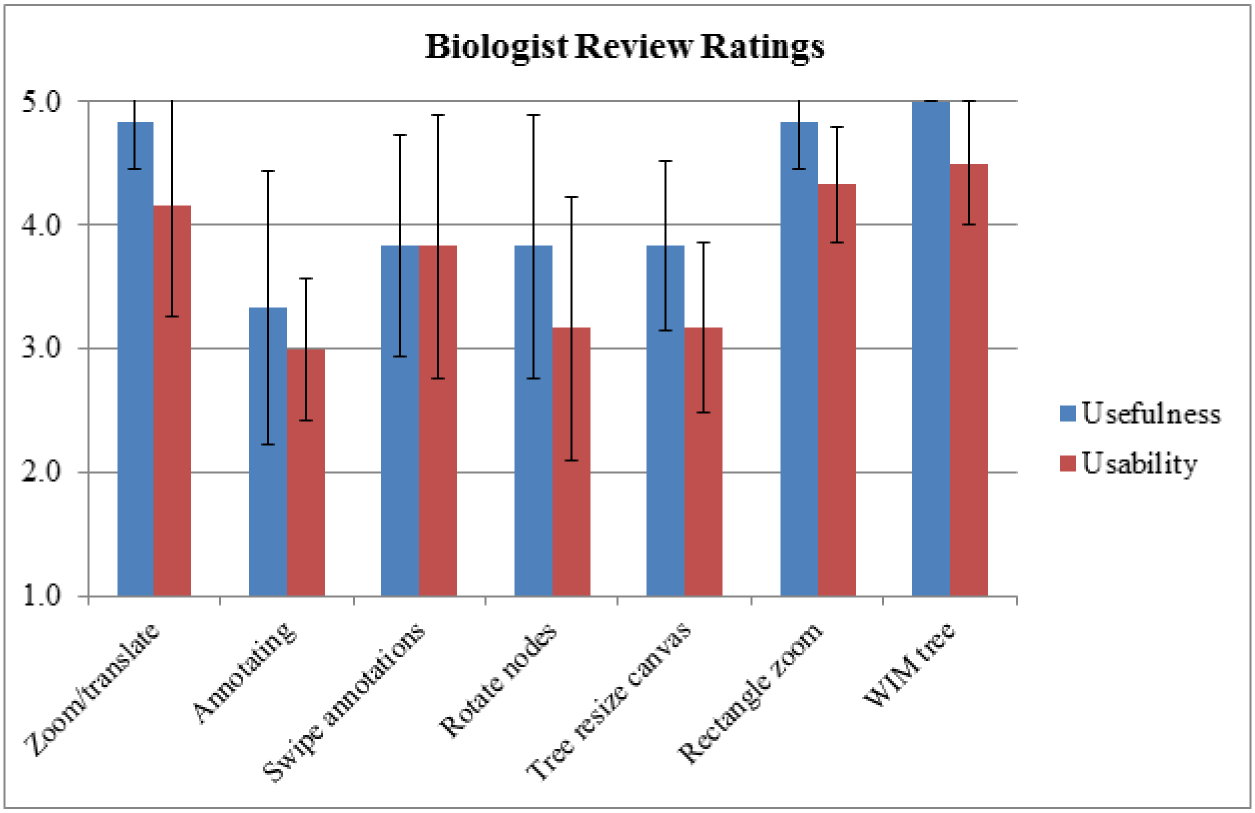


Figure 14: The average Likert ratings on features of PhyloPen, where the blue bar represents usefulness and the red represents usability.

Observing the results illustrated by Figure 14, there seemed to be a correlation between the usefulness and usability ratings of features. Usability was almost always slightly lower than the usefulness. This may have been due to shortcomings in the interface that caused the users to become frustrated. This is especially apparent when looking at the gap between usefulness and usability of navigation gestures, where pan and pinch touch gestures are supposed to be the most reliable to identify. As explained in participant comments about their personal experiences, there were times when using WPF that the touch gesture recognition would shut down for no apparent reason or become very unstable and hard to manage. It would happen at random intervals, requiring the user to reopen the application. The touch gestures also may not have operated as smoothly in general as we desired. The results do indicate, however, from the relatively low standard deviation, that the navigation gestures showed themselves (when working) to be highly useful across the board of participants.

Following the same pattern of having slightly less usability than usefulness, node rotation (clade child swapping) and the tree resize canvas also proved to have certain shortcomings in design. During this user evaluation, the node rotation worked by flipping the order of the two children instantaneously after recognizing the flipping gesture. This, however, can lead to context loss, especially in larger trees, leaving the user confused on the result of their actions. We hypothesize that adding visual transitions between the two states of the phylogenetic tree would have increased the usability and possibly the perceived usefulness as well. There was also some potential confusion in the design of the tree resize canvas. While the tree resize canvas operated on the standard Euclidean coordinate system, the graphical coordinate system used an inverted-y system. Thus, as the user moved the pen upward, the tree would grow in the opposite direction, creating a discrepancy in the visual intuition of the design. If the tree resize canvas followed the same inverted-y coordinate system, we hypothesize the user would have found more usefulness and usability in the feature.

The feature that scored the lowest in terms of both usefulness and usability was annotation. In reconsidering the system, we personally found the mechanism could have been too convoluted and complicated to be useful or enjoyable. In any case, there was certain evidence in the responses of the participants that hinted at some frustration with getting the annotation system to work. One potential point of frustration was having to double tap a tiny node circle with a pen to associate the annotation with it. Also, there was some difficulty activating gesture mode with the "X" gesture. In this version of PhyloPen, after drawing a line, it would disappear after a certain amount of time. When drawing the "X," by the time the user drew the second line for the symbol, the first line sometimes disappeared, requiring them to race to get the gesture drawn in time without too much error in the line. Some participants also expressed a preference for text annotations over ink annotations. In this version, we also passed copies of the same annotation up or down the tree. However, there are times the user would not have wanted to associate the annotation with all children when passing the annotation down. For example, the user may want to pass down a "can fly" annotation through the bird clades but wants to exclude it from chickens, ostriches, and other flightless birds. The original annotation was the same annotation, but we made copies of it. This is what led to the motivation of ultimately switching annotations from node-based to region-based.

Despite these shortcomings in the user interface design, the feedback from the participating biologists was overall positive. After watching the video demonstration, the participants were generally able to learn how to use the interface appropriately. A shortlist of the gestures was requested by a few participants for memory of the gestures, so it might be a good idea to use one in future experiments. Additionally, new features were suggested, the most requested of which was to be able to move a clade within a tree by re-attaching it to another ancestor.

In this chapter, we discussed our earliest work experimenting with various input modalities, ultimately focusing on pen and touch interfaces. After developing a prototype of PhyloPen we evaluated its usefulness and usability in a small prototype user study that showed overall positive results. In the next chapter, we discuss how feedback from this and consecutive prototype evaluations led toward a final design.

CHAPTER FOUR: MOVING TOWARDS A FINAL DESIGN

4.1 First Revision

In moving forward from our preliminary work, we wanted to create a new pen and touch system, particularly one that did not suffer from the limitations of WPF. When we previously developed a tool for Windows users only, in the user feedback, we were made aware that most experts in the field who would likely use the tool would also be using a Macintosh computer. There are certain biology related programs that are readily available for a Macintosh computer, because of its Unix underpinnings, that are not as common for Windows, posing a possible hindrance for whether PhyloPen could ever be put into broad use. Additionally, as previously noted, we were having issues with WPF's touch API in that it would mysteriously stop receiving touch events at random points in time, creating user frustration. Therefore, we investigated building PhyloPen from the ground up in a new platform/API.

While we would have liked to move PhyloPen to web integration, at the time of this writing, current web browsers do not expose built-in support for pen gestures. Wacom does offer a web browser plugin for support of their tablets, but we did not want to be limited to an external Wacom tablet. Surprisingly, there were many programming API's that provided support for touch but few had developed any pen/stylus features. In fact, the new foundation we chose, JavaFX, for its platform independence and multitude of similarities to WPF as a specialized user interface creation API, does not inherently support pen events. We had to integrate a pen support Java library into JavaFX called JPen. The developer of the library did not create built-in support for its use with JavaFX. However, they did provide a general means to integrate it. While WPF comes with an ink canvas class, we created our own in JavaFX to receive events from JPen. Like WPF, JavaFX also uses vector-based graphics, but unlike our previous experiences with WPF, the touch support seemed much smoother in general.

In moving from the old WPF system to the new JavaFX system, we made several changes to address issues and requests that were brought to our attention in the previous user study and additional feedback from biologists. The simplest was inverting the y-coordinate on the tree resize canvas, as shown in Figure 15 and based on user feedback, making the direction of size change of the tree the same direction that the user draws on the canvas to provide a more intuitive interaction.

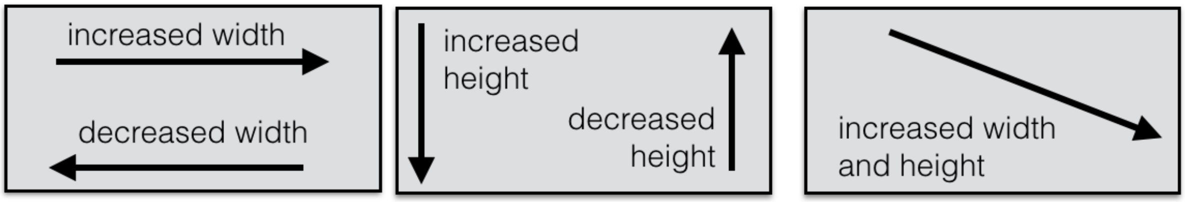


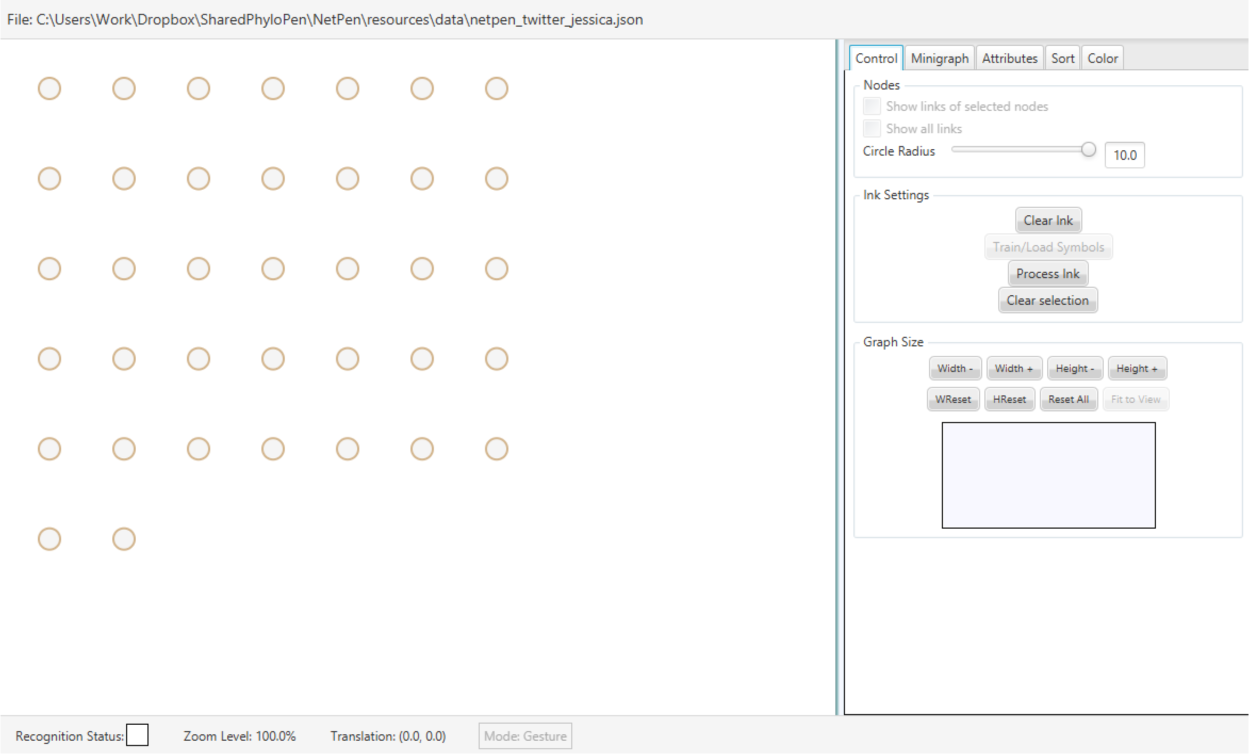
Figure 15: The revised tree resize canvas with an inverted y-coordinate system.

Changes were needed in how other gestures were processed as well. Previously, the user would make a one-line ink gesture and upon removing the pen, the gesture would be processed, leaving little room for user error. They would also have to be in the designated gesture mode to process the ink. Our new solution integrated both the navigation and gesture modes together by allowing the user to choose when to process the ink gesture, allowing the user to make corrections and evaluate their work as well as making navigation more seamless. In this manner, the user can write with one hand and navigate with the other at the same time. When the user was done writing, they could signal the program to process the ink by single-tapping the screen with the pen. This gesture was chosen due to its distinctiveness from other gestures and its minimal disruptiveness.

There was an additional reason for switching to making the processing of ink signals explicit. The previous system could only handle one-line gestures, but the new system allows for multiple lines in one gesture, providing more freedom in gesture creation. The new system allowed us to replace the loop gesture for expanding and collapsing clades to something more intuitive: a plus sign for expanding and a minus sign for collapsing. The plus or minus sign must be within some threshold distance from the center of the node for it to be registered with that node. In the case that two nodes are close to one another, the node to which the plus/minus sign is closest will be the one that gets expanded/collapsed. Additionally, for experimenting, drawing a triangle around the node would have the same result. The symbol could be interpreted as a delta symbol for change. In adding these new gestures, we also fixed how the expanding and collapsing worked. The expanding and collapsing of clades used to leave large gaps without the tree resizing itself, taking up the same amount of space as before it was collapsed but with additional whitespace. We improved it so that the tree repositions its nodes accordingly to reduce excess whitespace.

In this new gesture system, we also added a gesture for the feature requested most by users: the ability to cut a clade from a part of the tree and re-attach it to another ancestor. The user must draw two lines, one line through the branch the user wants to cut and the other line connecting from the clade's node (moving outward) that will become the new parent of the severed clade. The two lines should generally be about perpendicular in angle from one another although they can be slightly off angle and still be recognized acceptably.

One of the other major parts of the gesture system we had to address was the annotation mode. In doing so, we had to introduce a completely new gesture feature, selection. The feature, however, did not start in PhyloPen but in an experimental branch-off version of the same concept applied to social networks that we call "NetPen." We used connections in Twitter data as a sample data set. All Twitter messages are listed as connections (graph edges) between friends/followers, which are the nodes. Unlike PhyloPen, the tree is laid out in a rectangular grid of nodes as shown in Figure 16.



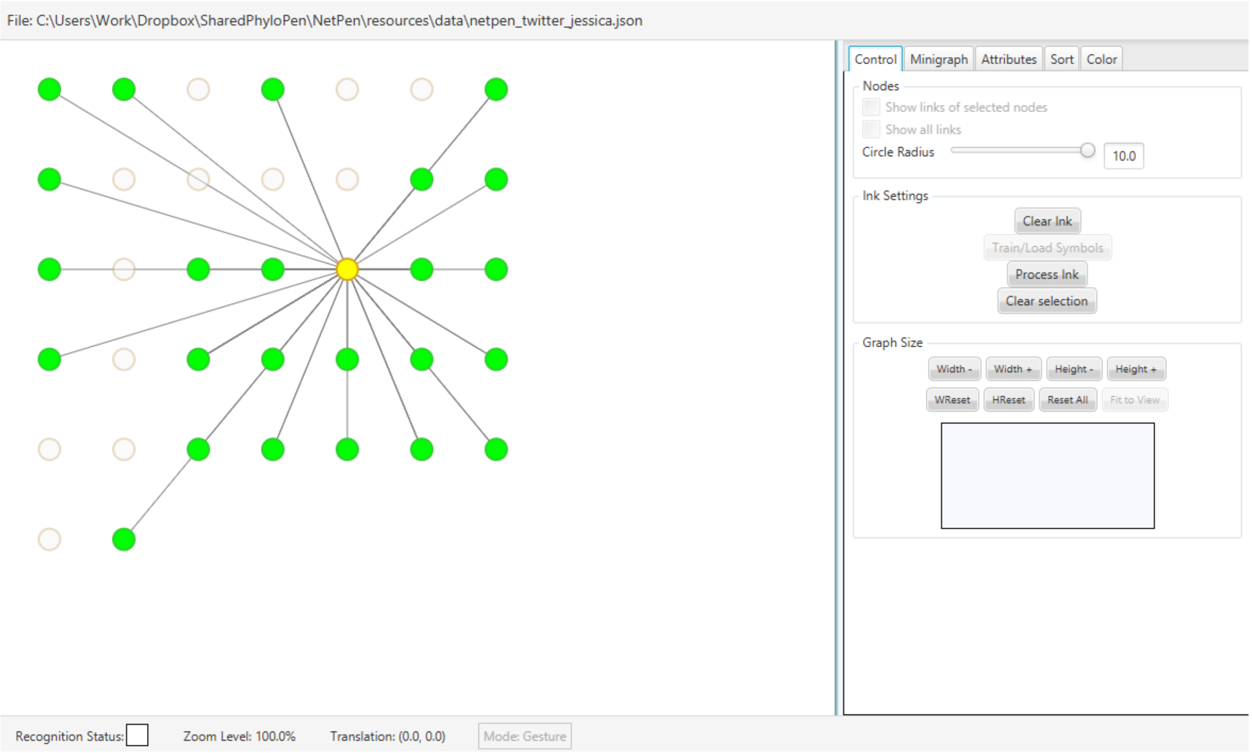
File: C:\Users\Work\Dropbox\SharedPhyloPen\NetPen\resources\data\netpen_twitter_jessica.json

Circle Radius 10.0

Recognition Status: □ Zoom Level: 100.0% Translation: (0.0, 0.0) Mode: Gesture

Figure 16: NetPen's tree layout is represented a rectangular grid of nodes.

Due to the high number of possible connections, none of them are visible by default to avoid confusion. To see the connections, the user must hover over a node, and its edges will be displayed, as shown in Figure 17.



File: C:\Users\Work\Dropbox\SharedPhyloPen\NetPen\resources\data\netpen_twitter_jessica.json

Circle Radius 10.0

Recognition Status:

Zoom Level: 100.0%

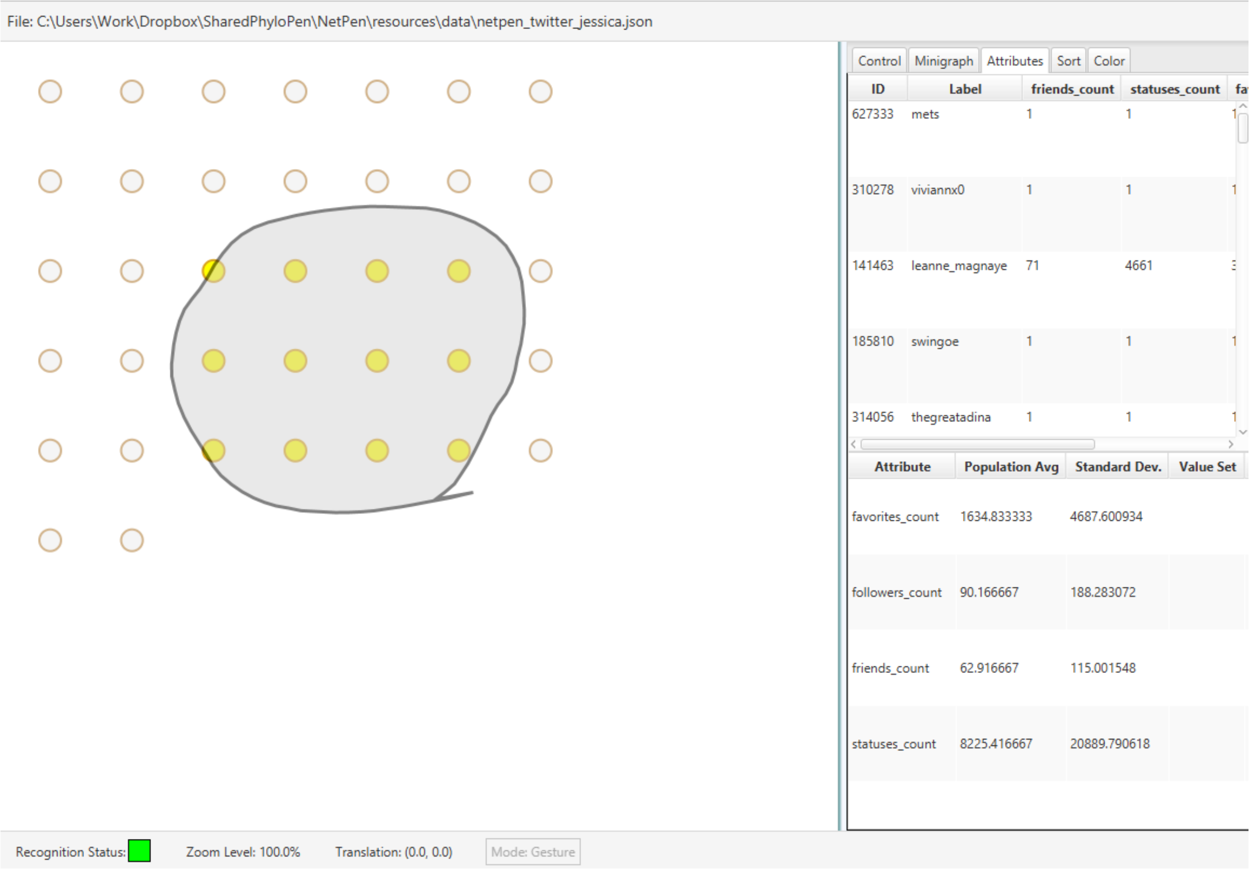
Translation: (0.0, 0.0)

Mode: Gesture

Figure 17: When hovering over a node in NetPen, the node's connections are displayed

Only the edges directly from the hovered node to its children are shown. As seen in the figure, if all the edges were displayed at once, it would be visually difficult in some cases to determine that all the lines were just edges from the one source node in yellow to all the green target nodes.

Selection in "NetPen" was developed for one purpose, to be able to display data about a subset of the population of nodes. The user can select a subset of the nodes by drawing an enclosed shape around them, as shown in Figure 18.



File: C:\Users\Work\Dropbox\SharedPhyloPen\NetPen\resources\data\netpen_twitter_jessica.json

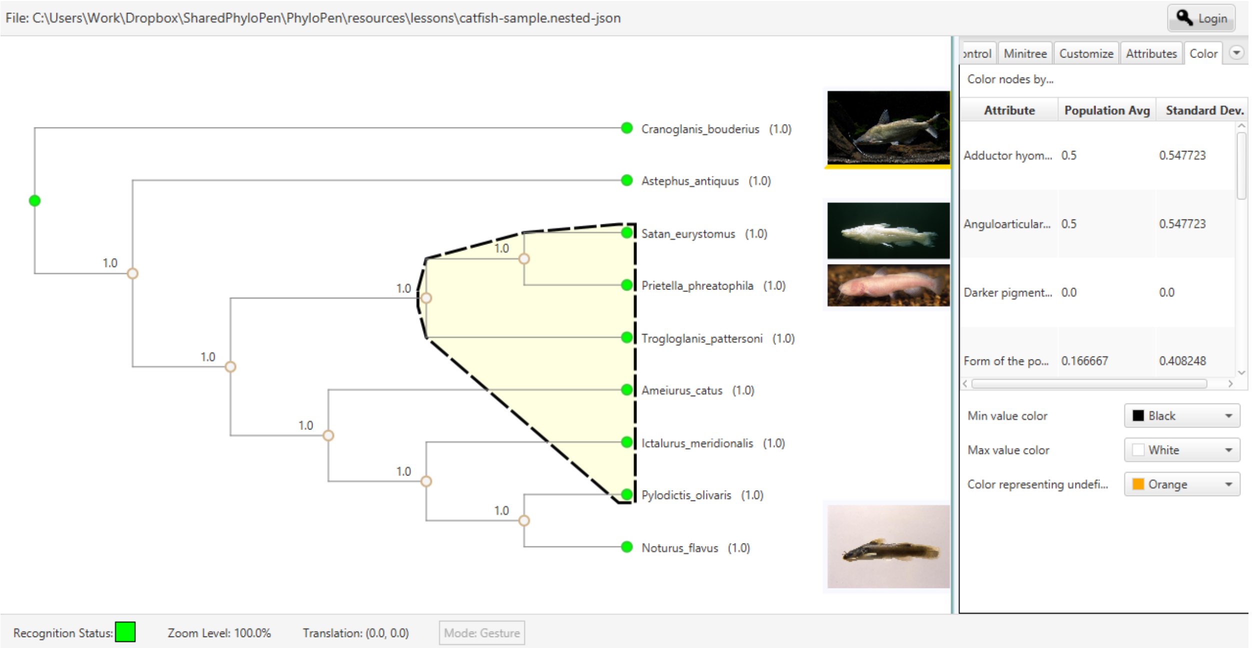
ID	Label	friends_count	statuses_count	fa
627333	mets	1	1	1
310278	viviannx0	1	1	1
141463	leanne_magnaye	71	4661	3
185810	swingoe	1	1	1
314056	thegreatadina	1	1	1

Attribute	Population Avg	Standard Dev.	Value Set
favorites_count	1634.833333	4687.600934	
followers_count	90.166667	188.283072	
friends_count	62.916667	115.001548	
statuses_count	8225.416667	20889.790618	

Recognition Status: 🟩 Zoom Level: 100.0% Translation: (0.0, 0.0) Mode: Gesture

Figure 18: Drawing an enclosing shape around the nodes selects them. The selected nodes appear in yellow.

All the selected nodes turn yellow, making it clear to the user which nodes are registered as selected. In the bottom table on the sidebar, the average and standard deviation of each attribute of the nodes are taken from the selected subpopulation. If no nodes are selected, the table displays the average and standard deviation of the attributes for the full population. We ultimately adapted this same feature to PhyloPen, including the subpopulation attribute averages table, but instead of leaving the ink circle behind and changing the color of the nodes, the nodes and involved branches are enclosed in a convex hull, as shown in Figure 19.


File: C:\Users\Work\Dropbox\SharedPhyloPen\PhyloPen\resources\lessons\catfish-sample.nested-json

Color nodes by...

Attribute	Population Avg	Standard Dev.
Adductor hyom...	0.5	0.547723
Anguloarticular...	0.5	0.547723
Darker pigment...	0.0	0.0
Form of the po...	0.166667	0.408248

Min value color: Black

Max value color: White

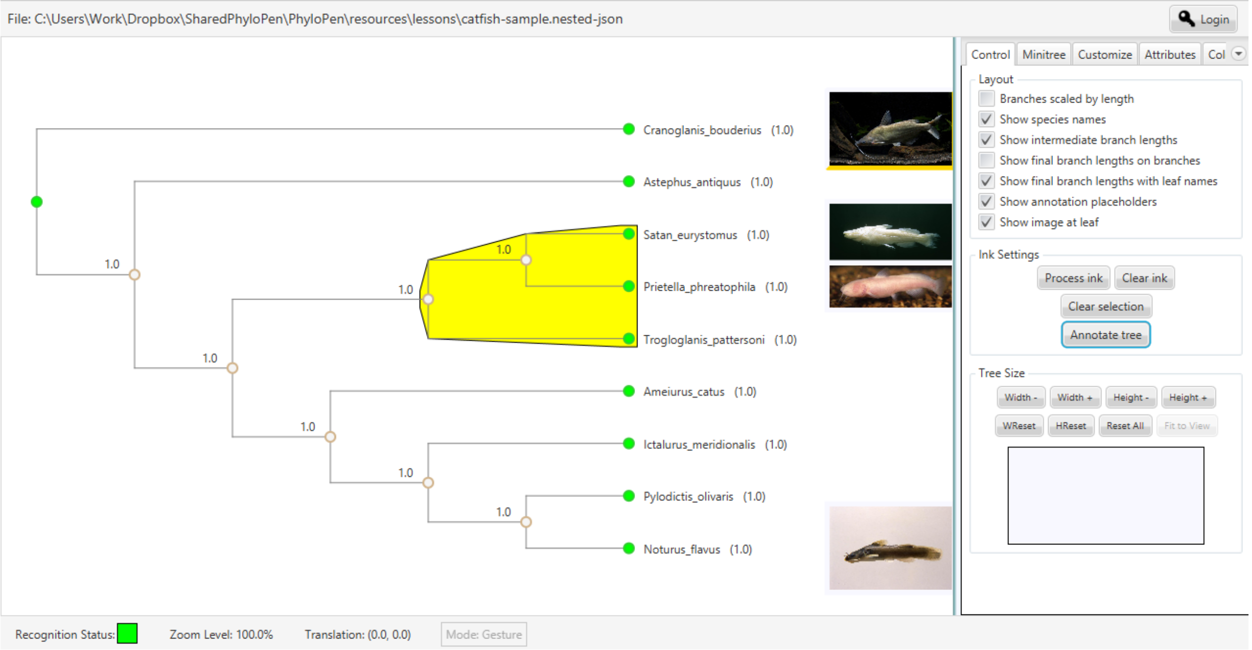
Color representing undefi...: Orange

Recognition Status: 🟩 Zoom Level: 100.0% Translation: (0.0, 0.0) Mode: Gesture

Figure 19: When the nodes are selected in PhyloPen, a convex hull encloses them.

A blinking fade in and out animation and dashed lines are applied to the convex hull to emphasize it is only a temporary visual. If the user taps anywhere on the canvas, the selection will disappear.

As for annotating in this new version of PhyloPen with the selection gesture, the user first enabled annotation mode by clicking the button labeled “Annotate tree” on the sidebar. This method of enabling annotation mode was a replacement for our arbitrary gesture choice we made in the previous version, but the use of an intuitive gesture to enable it is still open for study. Once in annotation mode, the user would select a group of nodes to annotate. Like normal selection, a convex hull is created around the selected nodes, but in this case, it has a solid color and shape, as shown in Figure 20.



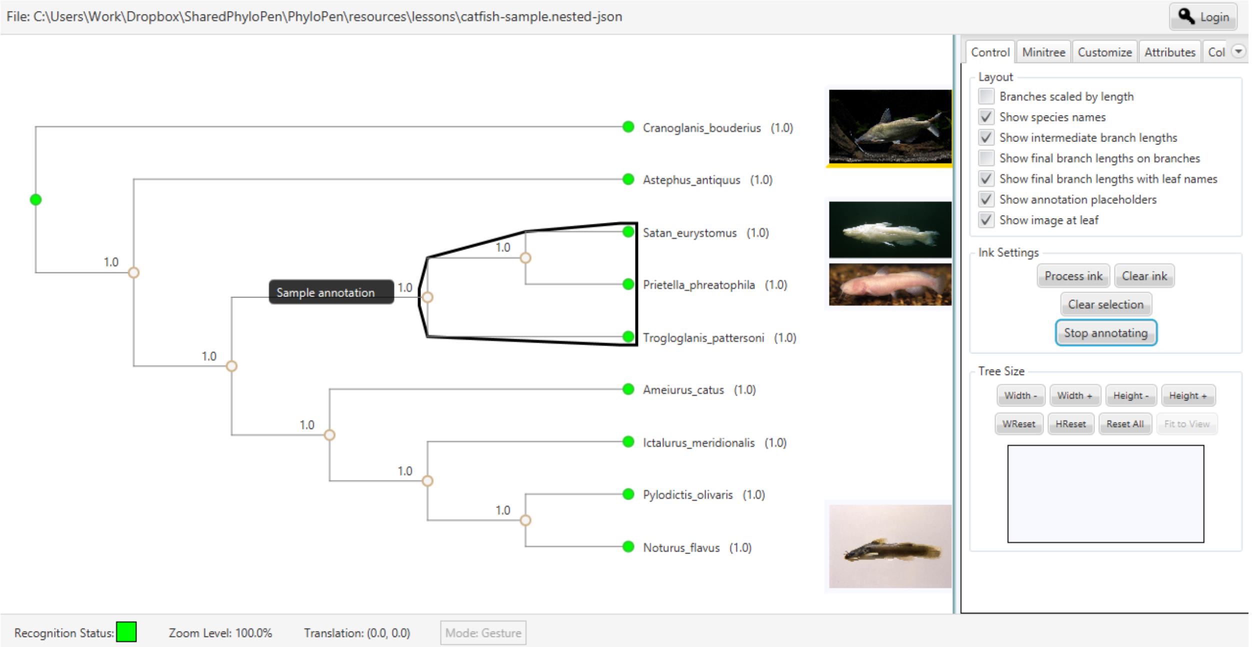
File: C:\Users\Work\Dropbox\SharedPhyloPen\PhyloPen\resources\lessons\catfish-sample.nested-json

Layout
- ☐ Branches scaled by length
- ☑ Show species names
- ☑ Show intermediate branch lengths
- ☐ Show final branch lengths on branches
- ☑ Show final branch lengths with leaf names
- ☑ Show annotation placeholders
- ☑ Show image at leaf

Recognition Status: 🟩 Zoom Level: 100.0% Translation: (0.0, 0.0) Mode: Gesture

Figure 20: When an annotation is associated with a group of nodes in the tree, a convex hull enclosing them acts as a placeholder for the annotation.

Upon selecting a group of nodes, the user would immediately be prompted with a dialog box to enter a text annotation. Ink annotations are not implemented but are a future possibility. The user could then see the annotation text by hovering over the convex hull, as shown in Figure 21.



File: C:\Users\Work\Dropbox\SharedPhyloPen\PhyloPen\resources\lessons\catfish-sample.nested-json

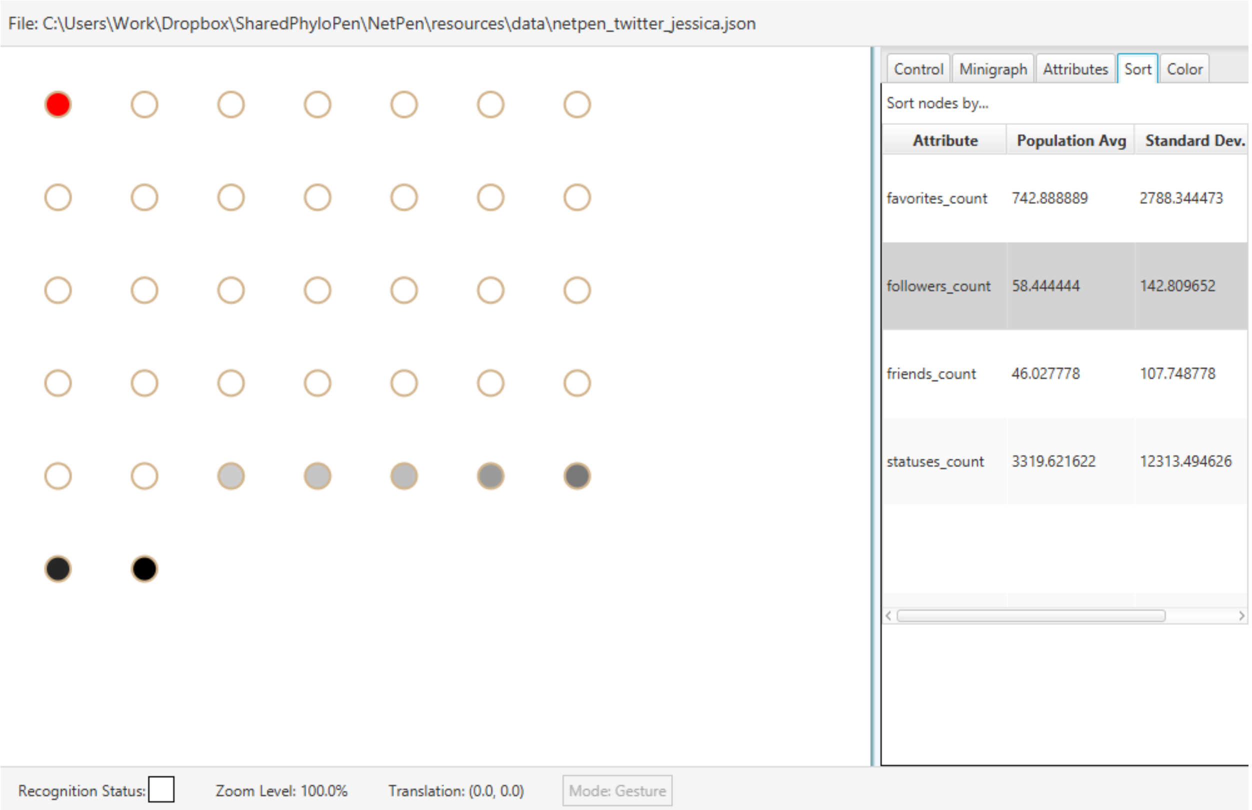
Layout
☐ Branches scaled by length
☑ Show species names
☑ Show intermediate branch lengths
☐ Show final branch lengths on branches
☑ Show final branch lengths with leaf names
☑ Show annotation placeholders
☑ Show image at leaf

Recognition Status: 🟩 Zoom Level: 100.0% Translation: (0.0, 0.0) Mode: Gesture

Figure 21: When hovering over the annotation's placeholder, the annotation text is displayed

To exit annotation mode, the user would click the button "Stop annotating." One important note to make about the change in the concept of annotation is that in the previous version, annotations were only associated with a single node. However, this concept is faulty because often the biologists would make notes about regions of the tree rather than just a single node. Therefore, the annotations are now region-based rather than node-based. These region-based annotations can be drawn around the parent node in the hierarchy as is traditional in phylogenetics or around the whole region, allowing for flexibility in choice of style.

The addition of features adapted from "NetPen" was not limited to selection. In "NetPen," we also implemented a means to sort and color nodes in the grid by an attribute's value. The relative intensity is calculated from a node's attribute value relative to the full population and then colored or sorted by that value. An example is shown in Figure 22.



File: C:\Users\Work\Dropbox\SharedPhyloPen\NetPen\resources\data\netpen_twitter_jessica.json

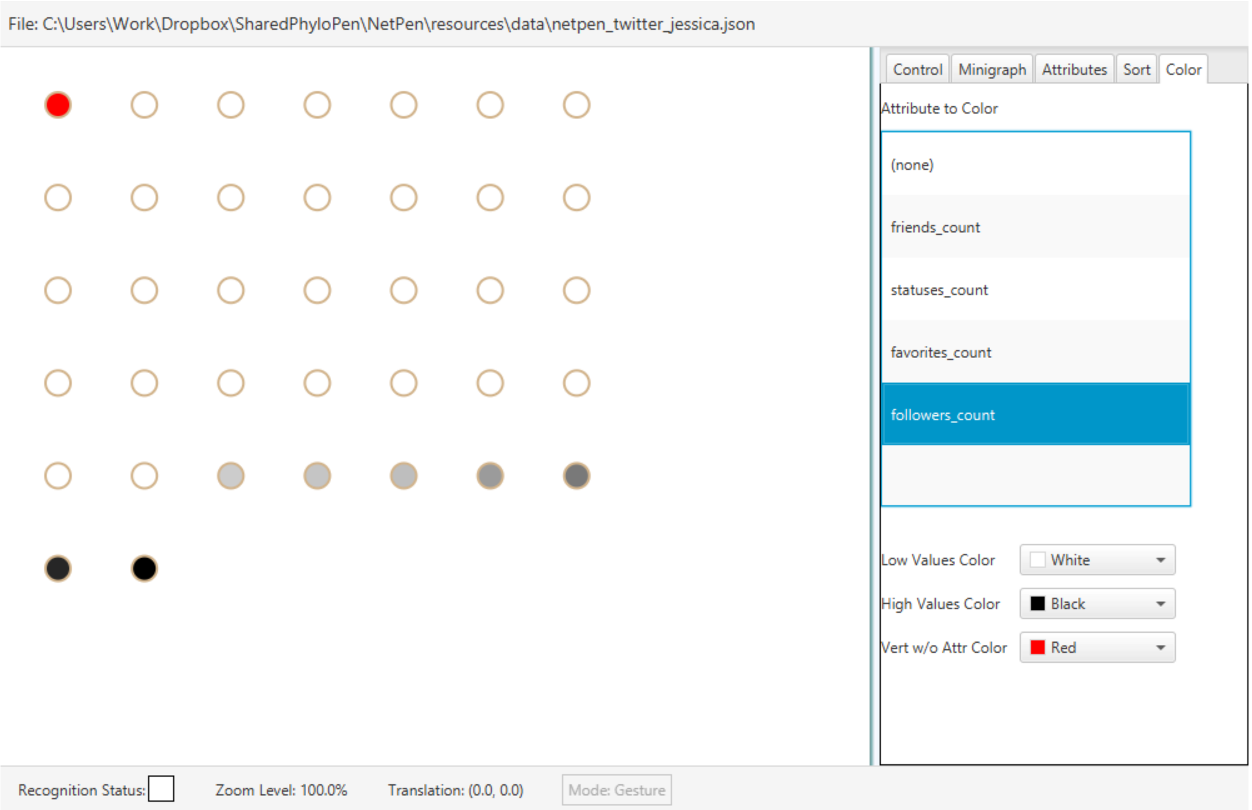
Sort nodes by...

Attribute	Population Avg	Standard Dev.
favorites_count	742.888889	2788.344473
followers_count	58.444444	142.809652
friends_count	46.027778	107.748778
statuses_count	3319.621622	12313.494626

Recognition Status: □ Zoom Level: 100.0% Translation: (0.0, 0.0)

Figure 22: The nodes in the grid can be colored by an attribute value.

The nodes in the figure are sorted and colored per the followers_count attribute, which can be accomplished by selecting the desired attribute in the table. The rows are intentionally made large for ease of touch. The colors used for the intensity range can be customized as shown in the "Color" sidebar tab in Figure 23.



File: C:\Users\Work\Dropbox\SharedPhyloPen\NetPen\resources\data\netpen_twitter_jessica.json

Attribute to Color

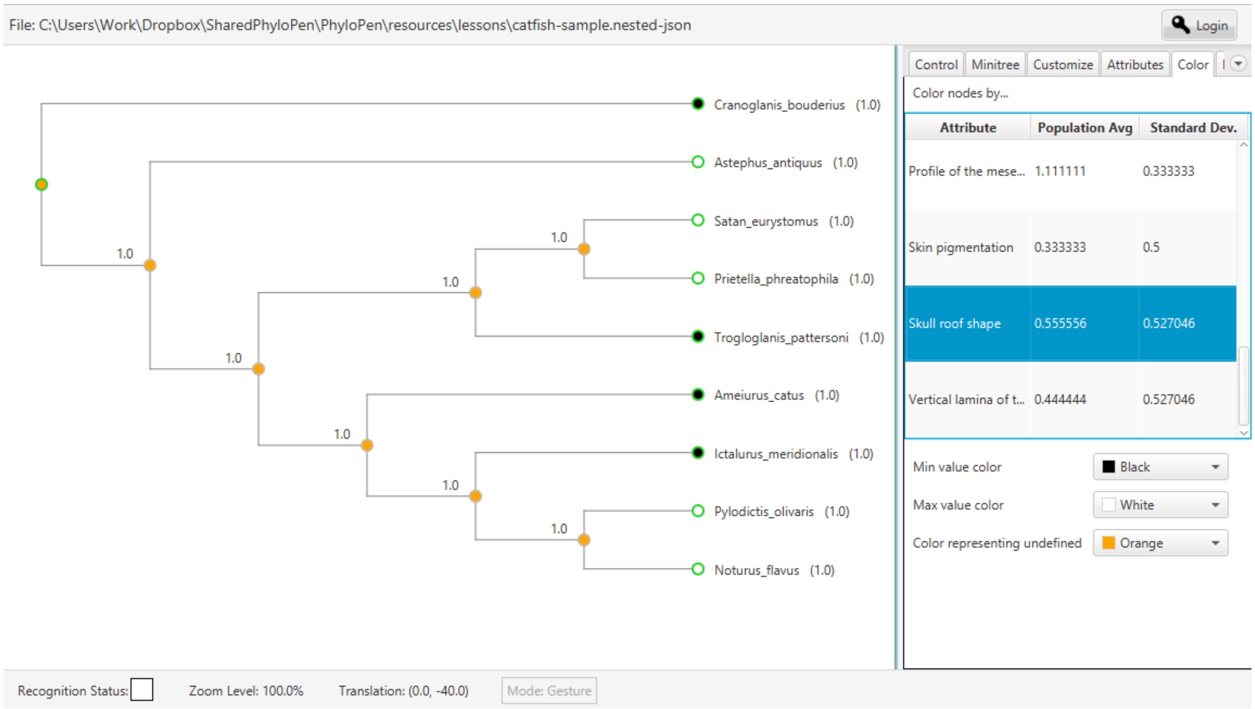
(none)
friends_count
statuses_count
favorites_count
followers_count

Low Values Color: White
High Values Color: Black
Vert w/o Attr Color: Red

Recognition Status: □ Zoom Level: 100.0% Translation: (0.0, 0.0)

Figure 23: The nodes in the grid can be colored by an attribute value. The user can select the color range they desire.

In this example, we use red to indicate nodes that do not have the selected attribute. Every other color is in the range between white (for the lowest value) and black (for the highest value). The color by attribute table has been added to PhyloPen and works the same as the one described in "NetPen," illustrated in Figure 24. We initially thought that the sorting feature did not apply to phylogenetic trees, but instead of sorting all the nodes in one line, the children of each clade could potentially be sorted in this fashion.



File: C:\Users\Work\Dropbox\SharedPhyloPen\PhyloPen\resources\lessons\catfish-sample.nested-json

Color nodes by...

Attribute	Population Avg	Standard Dev.
Profile of the mese...	1.111111	0.333333
Skin pigmentation	0.333333	0.5
Skull roof shape	0.555556	0.527046
Vertical lamina of t...	0.444444	0.527046

Min value color ■ Black

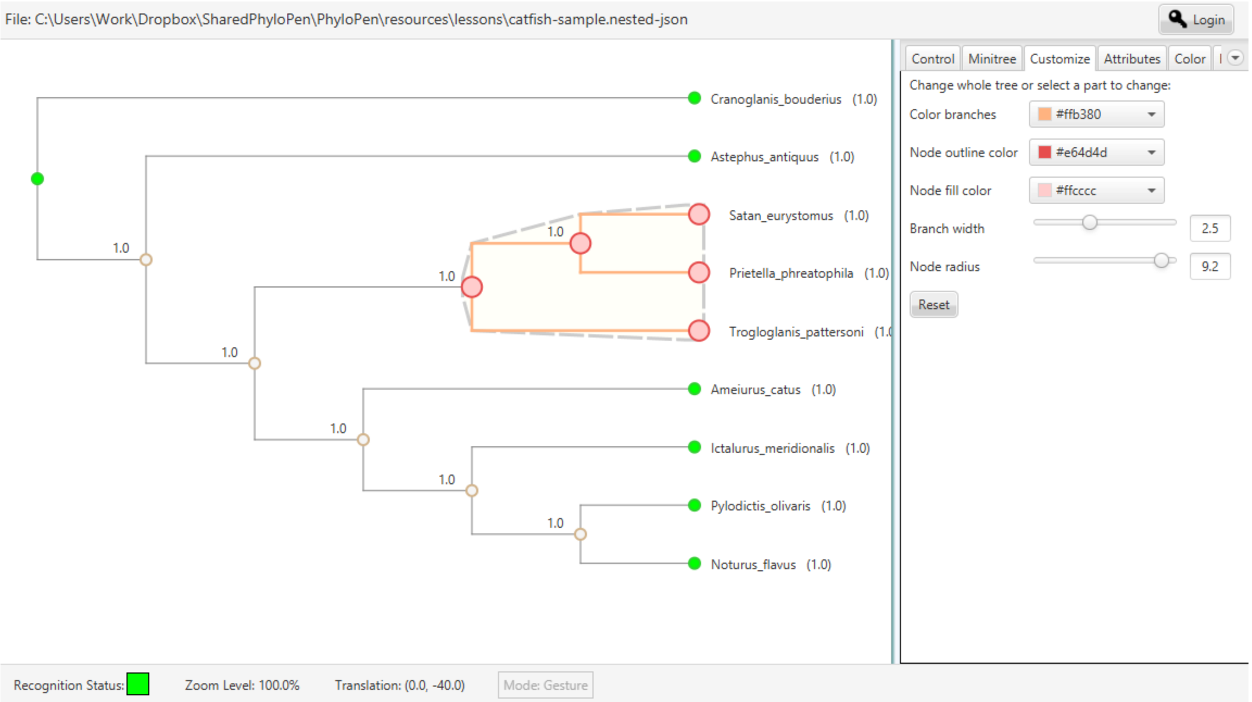
Max value color □ White

Color representing undefined 🟧 Orange

Recognition Status: □ Zoom Level: 100.0% Translation: (0.0, -40.0) Mode: Gesture

Figure 24: Adapted from NetPen, nodes can also be colored and sorted by an attribute value in PhyloPen.

Following from the node coloring and selection concepts, we additionally added the ability to customize tree appearance as highly requested by the biologists for the purposes of emphasizing points of the tree in the context of publishing and teaching. Once a selection is made with the selection ink gesture, the user can use any of the features in the "Customize" sidebar tab to change the appearance of the selected nodes and branches, as shown in Figure 25, including branch color, node outline color, node fill color, branch width, and node radius. There is additionally an option to reset the appearance to the default.



File: C:\Users\Work\Dropbox\SharedPhyloPen\PhyloPen\resources\lessons\catfish-sample.nested-json

Change whole tree or select a part to change:

Color branches: #ffb380

Node outline color: #e64d4d

Node fill color: #ffcccc

Branch width: 2.5

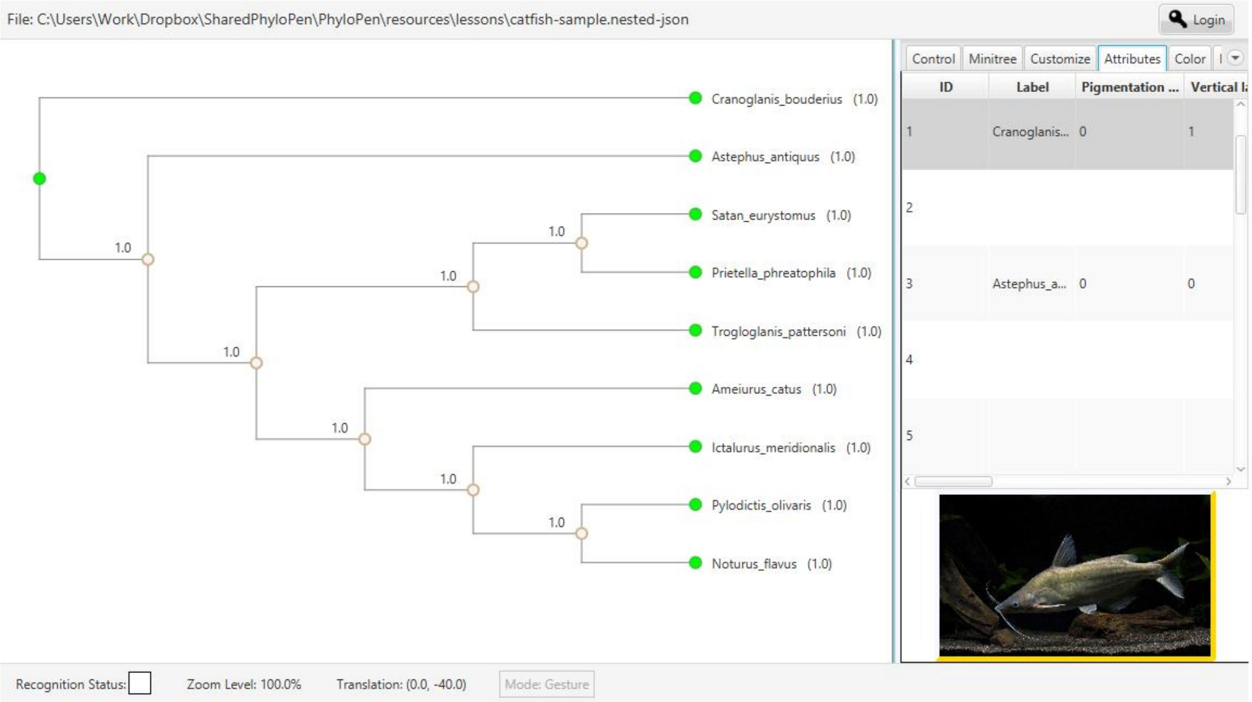
Node radius: 9.2

Recognition Status: 🟩 Zoom Level: 100.0% Translation: (0.0, -40.0) Mode: Gesture

Figure 25: Selecting a part of the tree allows the use to customize its appearance from the "Customize" sidebar tab.

In "NetPen," there was a new approach to displaying data, which inspired more changes in PhyloPen. When hovering over a node in the grid, the population averages for each attribute would temporarily show the values for only that one node. In our previous version of PhyloPen, hovering over a node would display a box of information that could potentially occlude other visual elements or could go off screen. In general, it was also inconvenient for the user to have to tap the node to keep the box open. Instead, displaying the information on the sidebar avoids this problem. While the averages table in PhyloPen does not change during hover, we instead select the row for the node in the table that lists attribute values for the whole tree. It remains selected even after the user moves away from the node by default, unlike the hover information box of the previous version.

It does not change until you hover over a different node. Selecting a row in the attribute table additionally displays the images for that node below the table as shown in Figure 26.



File: C:\Users\Work\Dropbox\SharedPhyloPen\PhyloPen\resources\lessons\catfish-sample.nested-json

ID	Label	Pigmentation ...	Vertical l
1	Cranoglanis...	0	1
2			
3	Astephus_a...	0	0
4			
5			

Recognition Status: □ Zoom Level: 100.0% Translation: (0.0, -40.0) Mode: Gesture

Figure 26: When hovering over a node, its corresponding row is selected in the attributes table, which displays information about its attributes and displays a picture representing the species.

The yellow border on the image indicates there is more than one image to be viewed for that node. Tapping on the box displays all the image previews (Figure 27).

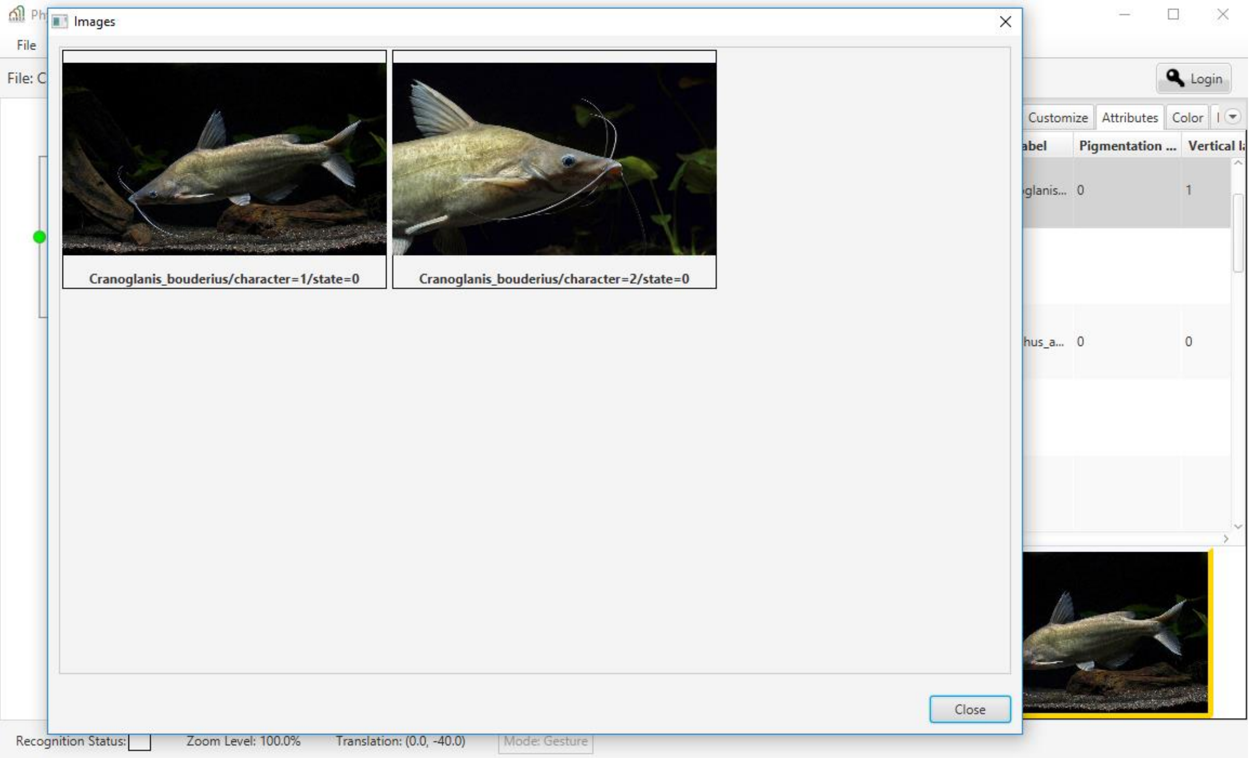
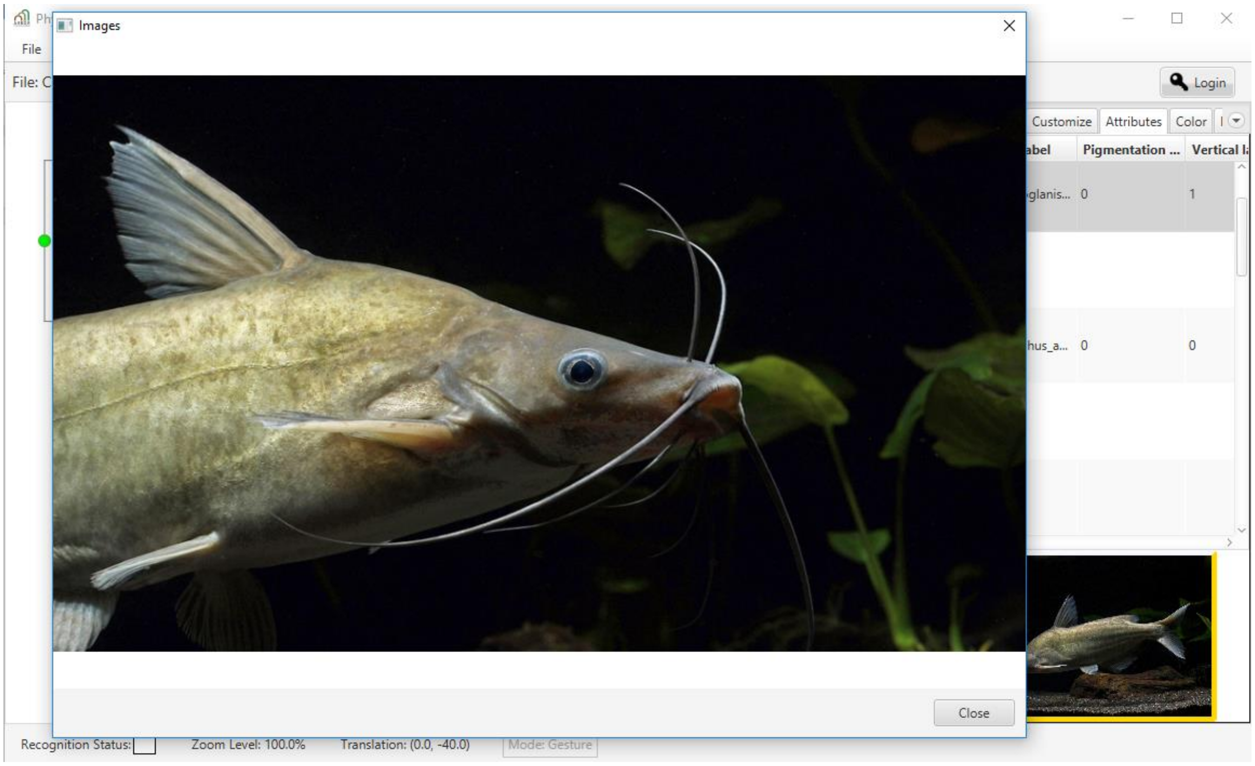


Figure 27: Tapping on the image under the attributes table displays previews of all the images associated with the node.

The user can then select any individual image for a more detailed viewing (Figure 28).



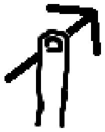
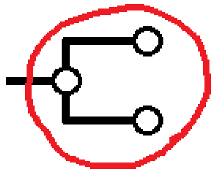
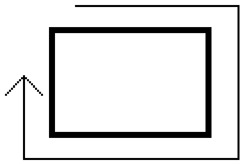
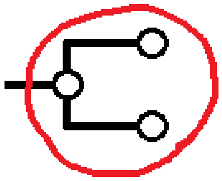
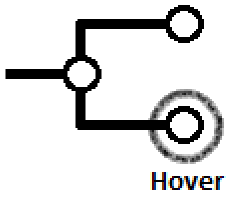
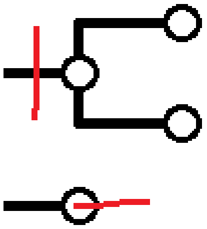
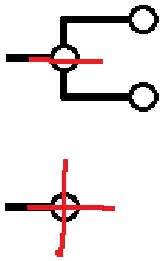
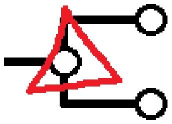
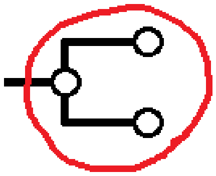
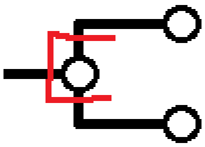
abel	Pigmentation ...	Vertical l
glanis...	0	1
hus_a...	0	0

Recognition Status: □ Zoom Level: 100.0% Translation: (0.0, -40.0) Mode: Gesture

Figure 28: Tapping the image previews will display the image in more detail.

As for other miscellaneous changes, we tried to increase the ink recognition accuracy to lower user frustration. While we had performed filtering through resampling the points previously, we tried adding a dehooking method devised by LaViola (2005) but still received unsatisfactory results. The ink recognition system needed to be expanded to something more complex than the heuristics we used to recognize shapes. In adding the selection gesture, we also changed the child node flipping gesture to a square bracket to differentiate it from any circular shape. Table 4 summarizes the overall gesture/feature set of the first JavaFX version of PhyloPen.

Table 4: The revised gestures and features of PhyloPen after the first prototype study.

Gesture	Name & Description	Gesture	Name & Description
	**Translation**: Swipe the screen with a finger.		**Zoom**: Pinch two fingers to zoom in and stretch them apart to zoom out.
	**Selection**: Draw an enclosed circular shape around a group of nodes and tap the canvas to select them.		**Zoom to Region**: Draw a rectangle around a region to gradually translate and zoom to that region as the new viewport and tap the canvas.
	**Annotation**: Once in annotation mode, tap the Annotate tree button, circle a group of nodes and tap the canvas to annotate. Tap the "Stop annotating" button to exit.		**Display Node Details:** Hover over a node with the pen, and information will be displayed in the Attributes tab of the sidebar.
	**Cut and Reattach**: Draw a line nearly perpendicular through the branch to cut, draw another line originating at the node where the clade should be reattached, and tap the canvas.	 ***Alternative method***: 	**Expand and Collapse Clades**: Draw a plus sign to expand a collapsed node-clade and a minus sign to collapse a node clade, tapping the canvas to process each. The symbols do not need to be perfectly positioned over the node, just within some proximity. ***Alternative method***: Draw a triangle around the clade-node to be expanded or collapsed and tap the canvas.
	**Customization**: Select the nodes/branches/clades with the selection gesture and customize their appearance with the Customize tab.		**Clade Child Swap**: Draw a square bracket symbol over the clade root and tap the canvas to swap its immediate child ordering.

4.2 Final Prototype Studies

After the first prototype study, two more were performed with biology students at the University of Central Florida. In the first study of the two, six undergraduate students were given two short exercises to complete taking turns with three devices over one class period with our revised version of PhyloPen. The first exercise's objective was to make sense of the phylogenetic tree presented to figure out which branch was wrongly placed in the diagram. Once the participant determined where it belonged, they would then move it to its proper place with the cut and reattach ink gesture. The second exercise was focused on marking up a phylogenetic tree with colors and annotations as an expert would do to emphasize properties of the tree. In the pre- and post-questionnaire, unlike the earlier prototype study, we included a short multiple-choice assessment to test their prior phylogenetic tree sensemaking skills and how it was affected after using PhyloPen. This assessment was adapted from The Tree-Thinking Challenge (Baum, Smith, & Donovan, 2005). Unfortunately, the results were largely the same before and after with little variation.

Due to the low number of students that were able to participate in the first round, we conducted a second study with twelve more students, giving them largely the same exercises, pre-questionnaire, and postquestionnaire. However, for the second round, we had the students sign up to participate outside of class time. Even with the lack of time constraint, the results before and after for the Tree-Thinking Challenge did not change. We concluded this was still due to the limited timespan they had to learn since the assessment tests more advanced concepts of interpreting trees than we gave the students the time to learn. Unless we were observing students over a period of weeks in class using PhyloPen as a teaching tool, it is unlikely we would have ever seen much change, making the assessment an unreliable tool for our final study.

Although these results did not achieve our desired results regarding tree sensemaking, the participants' observed and written experiences while using Table 4's version of PhyloPen were still informative enough to make revisions toward our final design. Two of the largest problems that caused participants confusion were the lack of ink gesture detection accuracy and having to explicitly tap the screen to process a gesture. The heuristics that assisted the template matching gesture recognizer were more of a hindrance. If one extra corner was detected, it could completely change what kind of shape the recognizer was expecting.

Based on the two prototype studies presented in this chapter, we uncovered changes to PhyloPen's user interaction paradigms that would make it more useful for the biology community and we realized a productive focus for our final study that would determine how well we were meeting the needs of biologists and would also inform future research in this area. Our final study presented in the next chapter is focused on the effectiveness of sensemaking that the tool has for the participant in relation to the standard tools that biologists currently use. In particular, it is a study of how PhyloPen stacks up against Mesquite, the discipline's current "gold" standard for tree visualization and manipulation.

CHAPTER FIVE: FINAL DESIGN AND STUDY

5.1 Final Software Design

To address the issues identified in our design during the final prototype studies, we focused on a solution that would address all the problems at once: a simpler ink gesture recognizer with a simpler ink gesture set. Instead of reverting back to processing gestures on a timer, with a simplified symbol set, we can determine from the first stroke drawn what are the possibilities following that stroke. If the first ink stroke of the gesture is a one-stroke gesture, we process immediately. If the first ink stroke contributes to a multi-stroke gesture, we wait for their next drawn stroke to process the whole drawn symbol. We can afford to create such a gesture set in this case due to the nature of the problem and its necessary operations. The changes are summarized in Table 5.

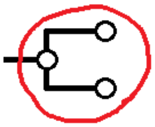
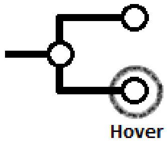
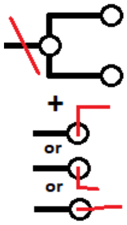
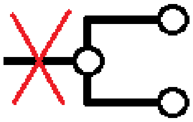
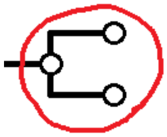
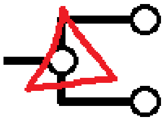
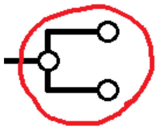
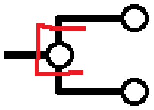
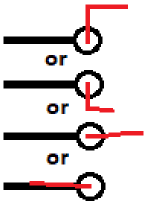
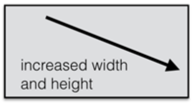
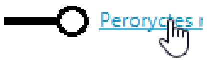
In this new gesture set, simply drawing a line could mean several possible operations, but it all depends on context how it is to be interpreted. If it is a line drawn vertically through a branch at a slant, it could be a cut and reattachment of a clade or a clade deletion. It depends on how the next stroke is drawn after the first line. If the second stroke is a proposed branch coming out of a node, that means cut the slashed branch and reattach at the node specified. If it is instead another vertical slash in the opposite direction (completing an X symbol), that means it is a deletion. If the first drawn line was just a branch coming out of the node, then that means we could add a new node and branch there. Alternatively, if the line is drawn backwards towards the parent, it means put a new clade/branch in between the node where the line starts and the node towards which the line is being drawn. As a note, adding new nodes and branches was an additional gesture we developed to be more comparable to the features of existing phylogenetic visualization tools.

If the first stroke does not make a simple line, it could be a one-stroke left square bracket to rotate the clade order or a one-stroke triangle (representing the delta symbol) to expand and collapse the clade. To make this simplified gesture set work, we unfortunately had to eliminate the plus and minus expand and collapse gestures. Any other shape that would not be detected by the template matching $N stroke recognizer with enough confidence would have to be a lasso selection, which is used for selection in customization and annotation and can be processed immediately.

In addition, while annotation required the user to enter annotation mode first by pushing a button in the previous gesture set, this is no longer necessary. The user can select the area to be annotated, press the annotate button, enter the annotation, and have a result without the risk of forgetting to leave annotation mode. The primary purpose of an annotation mode was to be able to perform a few different gestures to alter only the annotation. However, with the current gesture set, there is the potential to perform add and delete gestures on annotations with only the gestures already in the set by considering context alone. An X symbol does not necessarily only have to apply to deleting a clade. If we add an order of precedence in deletion, we can have a well-defined method to handle a deletion across multiple types of objects on the ink canvas. If we have an X over an annotation that overlaps a clade or its branches, the program will know to delete the annotation first since deleting a clade in an annotation would mean deleting the annotation by default.

Overall, with this new ink gesture recognition system (and set presented in Table 5), we have the potential for more gesture reuse and skipping the explicit processing step, hopefully reducing cognitive load and saving time, while also greatly increasing accuracy over our previous recognition system.

Table 5: The revised gestures and features of PhyloPen for our final design.

Gesture	Name & Description	Gesture	Name & Description
	Translation: Swipe the screen with a finger.		**Zoom**: Pinch two fingers to zoom in and stretch them apart to zoom out.
	**Selection**: Draw an enclosed circular shape around a group of nodes.		**Display Node Details:** Hover over a node with the pen, and information will be displayed in the Attributes tab of the sidebar.
	**Cut and Reattach**: Draw a vertical slash through the branch to cut, draw another line originating at the node where the clade should be reattached.		**Cut Clade (Delete)**: Draw an X through the clade's incoming branch to delete it.
	**Annotation**: Draw a lasso around a group of nodes to annotate, tap the Annotate tree button in the sidebar.		**Expand and Collapse Clades**: Draw a triangle around the clade marker to expand or collapse.
	**Customization**: Select the nodes/branches/clades with the selection gesture and customize their appearance with the Customize tab.		**Child Order Swap**: Draw a square bracket symbol over the clade root and tap the canvas to swap its immediate child ordering.
	**Add Node/Branch**: Draw an L or straight line branch from the node to create a new child. Additionally, drawing a straight line backwards from the node will re-parent the clade with a new clade.		**Tree Resize Canvas**: Draw a line on the canvas in the bottom right corner. The tree size change will be proportionate to the direction and length of the line you draw. Note: The tree grows downward.
	**Quick Web Info**: Tap on the name of the species to be directed to a webpage with more information about them.		

5.2 Final Study Objective

Having a final software design fully implemented, we needed a final summative study to properly measure its impact on the field. More precisely, our objective in the study reported in this chapter is to compare the interactive visualization pen-and-touch application, PhyloPen, against the state-of-the-art in present-day use by biologists, Mesquite, in their effectiveness for sensemaking of phylogenetic trees. We determine effectiveness using several dependent measures:

1. Completion time – The total time elapsed to complete all assigned exercise tasks.
2. Usefulness – A quantitative rating of perceived usefulness of the software.
3. Satisfaction – A quantitative rating of satisfaction with the software.
4. Ease of Learning – A quantitative rating of how easy it is to learn to use the software.
5. Ease of Use – A quantitative rating of how easy it is to use the software.
6. Cognitive load – A quantitative measure of cognitive load put on the user when using the software.
7. Task completion – The percentage of assigned exercise tasks that the user was able to complete.

The dependent measures of Usefulness, Satisfaction, Ease of Learning, and Ease of Use are the four dimensions of usability defined in a pre-validated survey called the USE Questionnaire (Lund, 2001). A series of 30 questions are provided that cover each of the dimensions and rate each response on a Likert scale from 1 (“Strongly Disagree”) to 7 (“Strongly Agree”). The responses are averaged by dimension. Separately, the dependent measure of cognitive load is the score given by completing a pre-validated survey called the NASA-TLX (Hart & Staveland, 1988). The participant is given six scales to rate their experience from 0 to 100 in terms of mental demand, physical demand, temporal demand, performance, effort, and frustration to calculate a final score on the same scale. The higher the final score, the larger the cognitive load.

To conduct a summative study, we established hypotheses to be tested for each dependent measure:

H1. Completion time – Users will be significantly faster at completing assigned tasks in PhyloPen than Mesquite.

H2. Usefulness – Users will find PhyloPen to be significantly more useful than Mesquite.

H3. Satisfaction – Users will find significantly more satisfaction in PhyloPen than Mesquite.

H4. Ease of Learning – Users will find PhyloPen and Mesquite to have equal difficulty for users to learn to use.

H5. Ease of Use – Users will find PhyloPen to be significantly easier to use than Mesquite.

H6. Cognitive load – Users will find PhyloPen to have a significantly lower cognitive load than Mesquite.

H7. Task completion – Users will be able to complete assigned tasks significantly more often using PhyloPen than Mesquite.

5.3 Final Study Procedure

To be able to compare Mesquite and PhyloPen fairly, we had to put some constraints on features to make the two as comparable as possible. For example, while PhyloPen has greater speed and space for rendering larger trees than Mesquite, which only uses the window space it has available to render the tree, using large trees for each would be too trivial of a comparison and would ultimately show nothing of interest. Thus, we limited our scope to relatively small phylogenetic trees. Mesquite is also a software system with a broader scope of purposes. We had to ignore the features that were not a part of PhyloPen and preset the workspace so that they could easily focus on only the tree rendering and character matrix editing aspects of the software. We also had to ignore some features that PhyloPen had that Mesquite did not, such as the ability to recolor nodes according to character matrix values and changing the width and height of the tree. These comparable features for Mesquite are outlined in Table 1. The table is intentionally meant to mirror the capabilities listed in Table 5.

In preparation for the study, we used G*Power to determine with a prior power analysis that, for a medium effect size (0.5) with a power of 0.80, 27 participants would be needed to validate the study. With that number in mind, we asked the biology department at the University of Central Florida for volunteers willing to participate. Our primary target of recruitment was an introductory evolutionary biology course, but general recruitment of biology undergraduates with some basic knowledge of phylogenetic trees was also made via email announcement and an online signup sheet. Each participant was promised a $20 Barnes and Nobles gift card for their time. They each participated one at a time using a Microsoft Surface Pro 2 tablet. A scroll wheel mouse was provided for exercises with Mesquite, and a stylus was provided for exercises with PhyloPen.

To begin a study session, the participant chose an ID randomly out of a pool of numbers that would identify them on results anonymously. They were then given an online prequestionnaire to collect information on their prior knowledge and demographics. Demographic information we asked for included age, gender, major/occupation, year in college. We also presented them with a series of questions to rate on a Likert scale 1 to 5:

1. How would you rate your level of knowledge in biology? (Beginner to expert)
2. How would you rate your level of knowledge in phylogenetics? (Beginner to expert)
3. How would you rate your general speed of learning new concepts? (Slow to fast learner)

4. How would you rate your level of experience with touch displays? (Little or no experience to extensive experience)
5. How would you rate your level of experience with electronic pen (stylus) devices? (Little or no experience to extensive experience).

We also asked for details on their prior experiences with other evolutionary biology software:

6. How would you rate your level of experience with the software system Mesquite?
7. How would you rate your level of prior experience with the software system PhyloPen?
8. How would you rate your level of prior experience with the software system FigTree?
9. How would you rate your level of prior experience with some other phylogenetic software?
10. List names for other phylogenetic software systems and/or packages you have used.

After completing the prequestionnaire, the participant was assigned either PhyloPen or Mesquite randomly with counter-balancing and given a command summary sheet for the corresponding software system. The first part was the training phase meant to help them become familiar with the software. In this phase, they could ask the proctor for help in learning how to use the different commands. They were given five exercise tasks that required the following:

1. Interpret the presented data to figure out which clade has the least amount of useful information and remove it.

2-1. Determine a misplaced part of the tree. The participants were specifically asked to find a clade that was labeled under the wrong family.

2-2. Move the misplaced clade around and restructure the tree correctly.

3. Color parts of the tree according to characteristics of the species.
4. Annotate according to those same characteristics of the species.

The evaluation phase uses these same exercise tasks but with a different randomly assigned dataset (with counter-balancing). The difference in this phase is that they must complete the tasks largely on their own and are timed while doing it for total completion time. They were also allowed to give up on a task if they became too frustrated. Once the evaluation phase was finished, the participant was given an online postquestionnaire containing questions from the USE Questionnaire and separately an online form for the NASA-TLX (Sharek, 2009). We used the shortened version of the NASA-TLX.

Once finished with the postquestionnaire for the first assigned software system, the participant then repeated the same process for the second assigned software system but with a different dataset from the first evaluation phase on the second evaluation phase. The postquestionnaire for the second software system included an extra final section that explicitly asked which system they preferred. The accompanying questions are as follows:

1. Which software system did you like more?
2. List three reasons for why you liked one over the other.
3. Which did a better job of presenting the data in a meaningful way and why?
4. Was there anything you liked about the other software over the one liked more in general?
5. What did you feel could be improved upon for either software application that wasn't already mentioned?

With these questions, we can have a more thorough understanding of the user's experiences and reasoning.

5.4 Final Study Results

In total, 27 undergraduate students participated in the study. All were biology majors except for one in environmental studies and one in health sciences, each with a background in biology. The distributions of gender and years as an undergraduate are shown in Figure 29, which shows most students being female and in their 4th year of college. Ages ranged from 19 to 48 years old with a mean of 22.63±5.76 years old. As shown in Figure 30, participants were mostly confident in their knowledge of biology with an intermediate to very knowledgeable rating but slightly less confident in their knowledge of phylogenetics with a somewhat to intermediate knowledgeable rating.

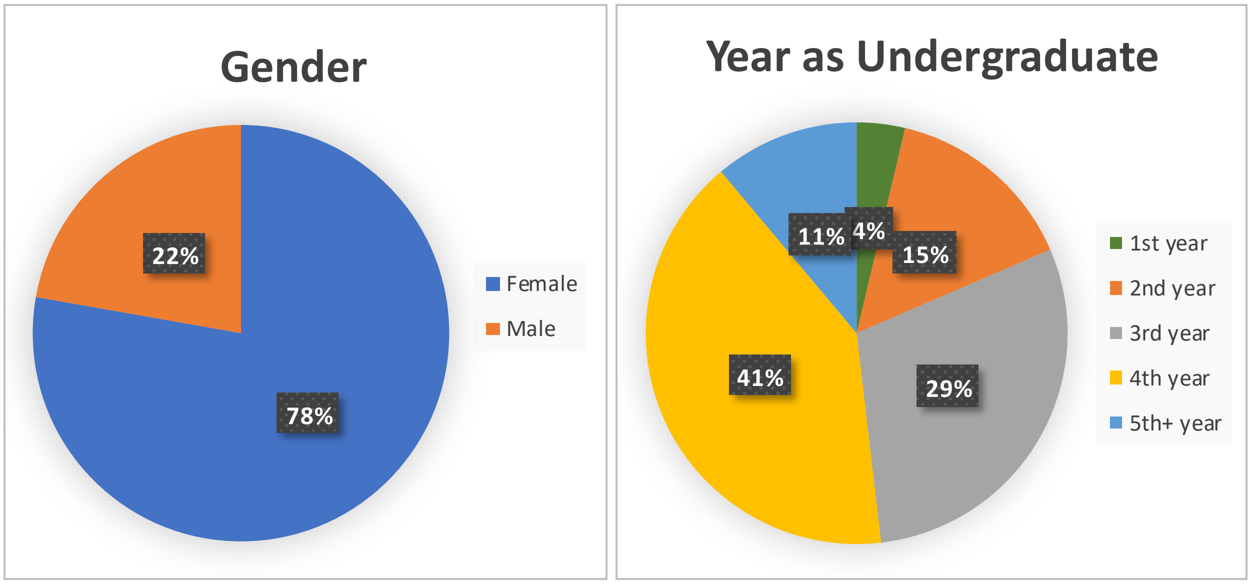


Figure 29: The gender and year distributions of students that participated in the final summative study.

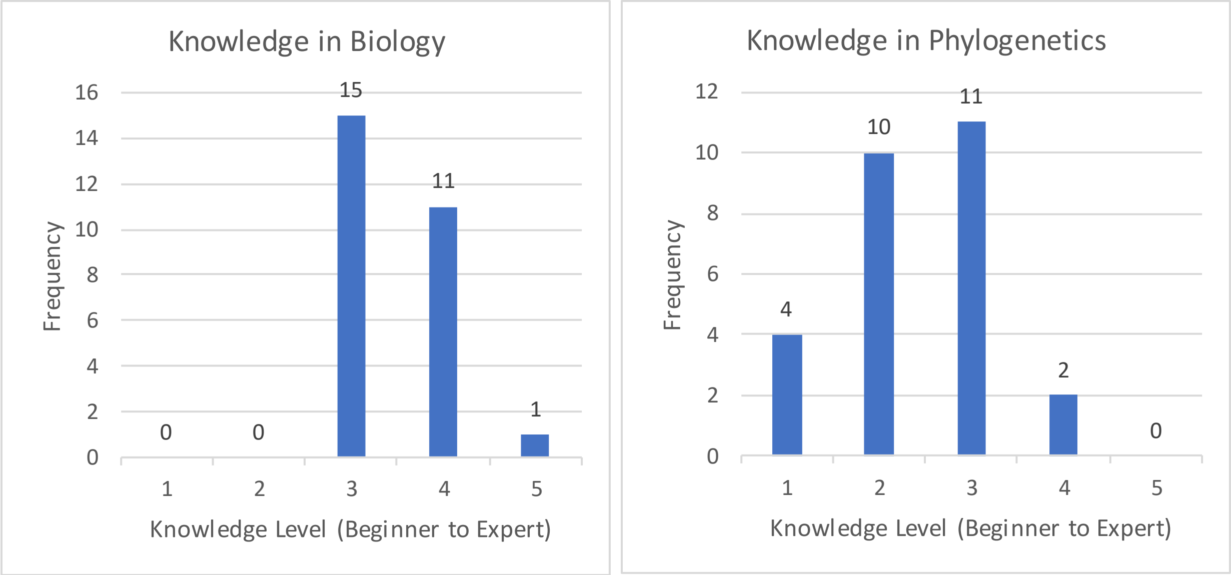


Figure 30: The self-rated knowledge distributions of the participants in biology and more specifically in phylogenetics.

Most also thought themselves to have a decent learning speed when presented with new concepts as shown in Figure 31. Reflected in today's common use of touch-based smartphone devices, Figure 32, ratings of prior touch experience, were high while stylus experience is mixed.

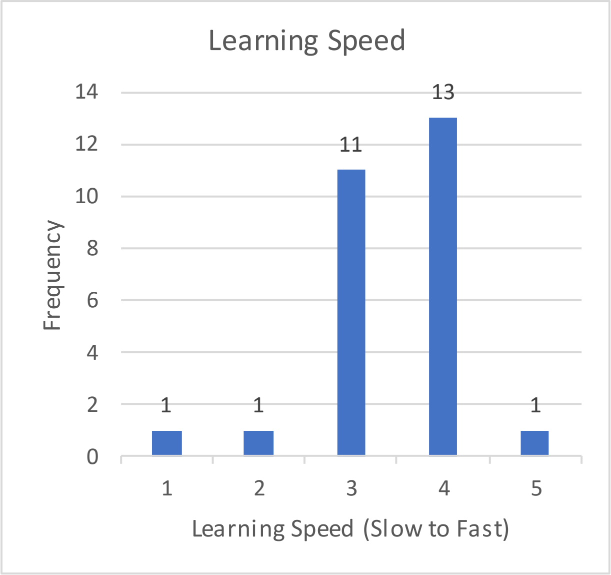


Figure 31: The self-rated learning speed distributions of the participants.

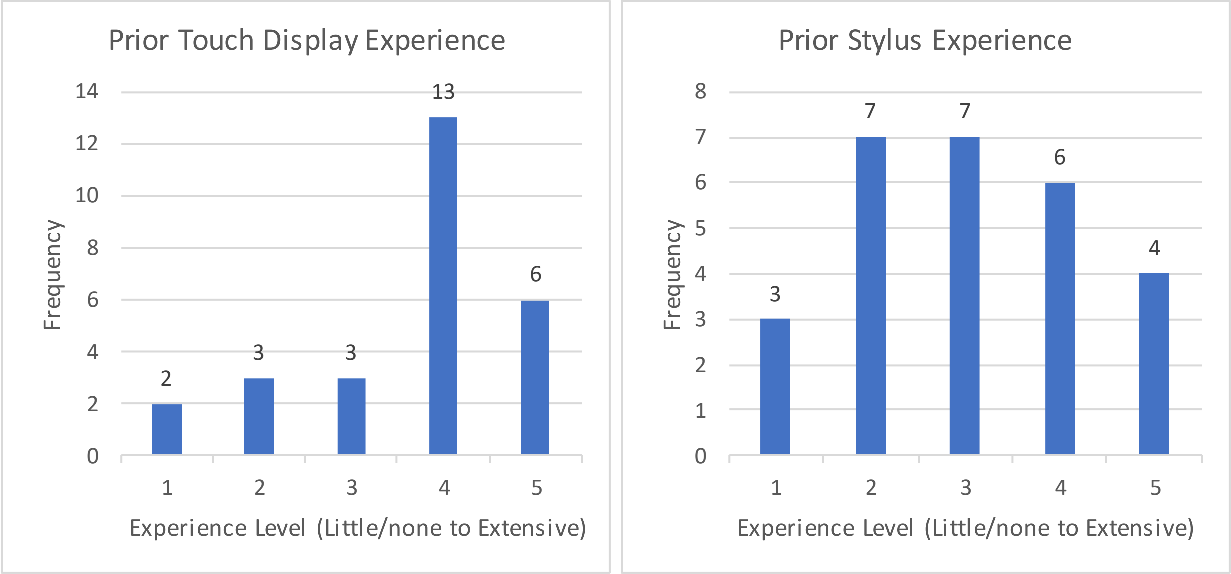


Figure 32: The distributions of self-rated prior experience with touch and stylus devices.

While stylus devices continue to make advancements, they still lag in popularity and technological advancement compared to touch functionality, which likely is the reason experience is more spread out for stylus devices.

To be able to use our scale-based measures, we first ensured their internal consistency, or reliability that they were measuring what was expected consistently. We calculated Cronbach's alpha for the USE Questionnaire and NASA-TLX dependent measures as shown in Table 6. Any value higher than 0.7 is considered to acceptable, and all measures met this criterion.

Table 6: Pre-analysis to confirm internal consistency of our dependent measures with Cronbach's alpha, and a Shapiro-Wilk to test the normality of each distribution.

Dependent Measure	Cronbach's Alpha	Shapiro-Wilk Test	Distribution
Completion time	N/A	$W = 0.96, p = 0.090 > 0.05$	Normal
Usefulness	0.75	$W = 0.88, p = 5.0 \times 10^{-5} < 0.05$	Non-normal
Satisfaction	0.97	$W = 0.89, p = 1.6 \times 10^{-4} < 0.05$	Non-normal
Ease of Learning	0.82	$W = 0.91, p = 6.2 \times 10^{-4} < 0.05$	Non-normal
Ease of Use	0.92	$W = 0.93, p = 3.8 \times 10^{-3} < 0.05$	Non-normal
Cognitive load	0.97	$W = 0.94, p = 0.013 < 0.05$	Non-normal

With internal consistency confirmed, we needed to examine the shape of the distributions for each measure to know which statistical measure would be appropriate to use for hypothesis testing. We used the Shapiro-Wilk Test to test the normality of the distributions as shown in Table 6. The results determined that completion time had the only normal distribution, so we plotted the distributions in Figure 33 to help determine the other shapes. As suggested by the Shapiro-Wilk Test, completion time appears approximately normally distributed between both treatments. The USE Questionnaire measures are left skewed, revealing more positive responses from participants than negative or neutral. The same applies to the NASA-TLX scores, except that lower scores are better for cognitive load, so it is right skewed instead. This suggests a positive bias in answering.

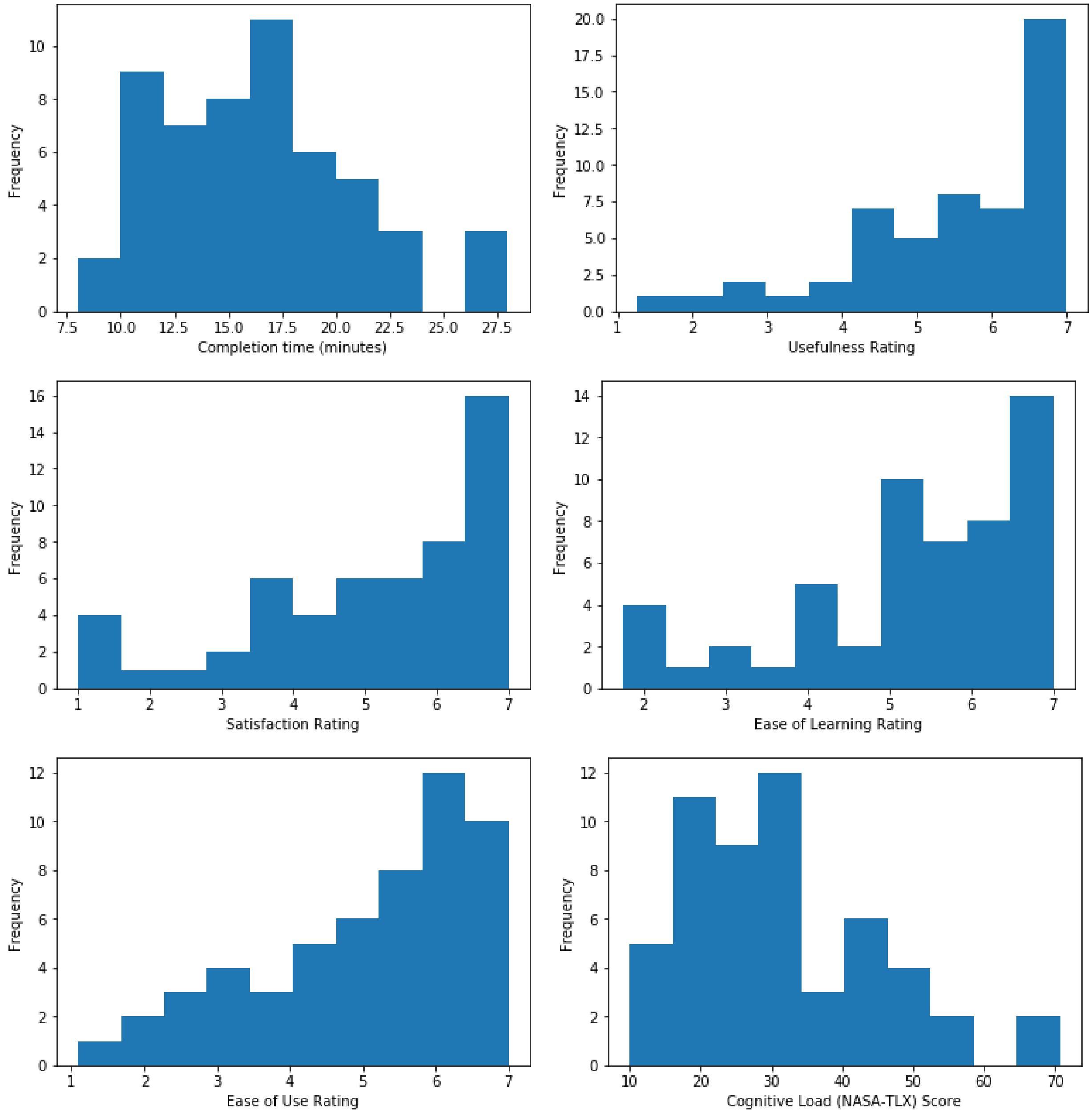


Figure 33: Distributions of the dependent measures for both PhyoPen and Mesquite.

With the shape of the distributions known, we determined that we could apply Student's paired t-tests to completion time and the Wilcoxon signed test to the other measures to determine whether there are statistically significant differences between each treatment. The results are shown in Table 7.

Table 7: Testing for statistically significant differences in each measure's distribution to confirm or reject hypotheses.

Dependent Measure	Test	Result	Interpretation
Completion time	Paired t-test	W = 0.70, $p = 0.49 > 0.05$	Same distributions
Usefulness	Wilcoxon signed test	W = 56.0, $p = 1.3 \times 10^{-3} < 0.05$	Different distributions
Satisfaction	Wilcoxon signed test	W = 24.5, $p = 5.5 \times 10^{-4} < 0.05$	Different distributions
Ease of Learning	Wilcoxon signed test	W = 49.0, $p = 1.3 \times 10^{-3} < 0.05$	Different distributions
Ease of Use	Wilcoxon signed test	W = 50.0, $p = 2.5 \times 10^{-3} < 0.05$	Different distributions
Cognitive load	Wilcoxon signed test	W = 72.0, $p = 4.5 \times 10^{-3} < 0.05$	Different distributions

The paired t-test determined that the variation in completion time between PhyloPen and Mesquite are relatively the same. We visualize this with a boxplot and histogram in Figure 34. While PhyloPen shows a slightly favorable range in completion time, it is not determined here to be significantly different. This result rejects H1.

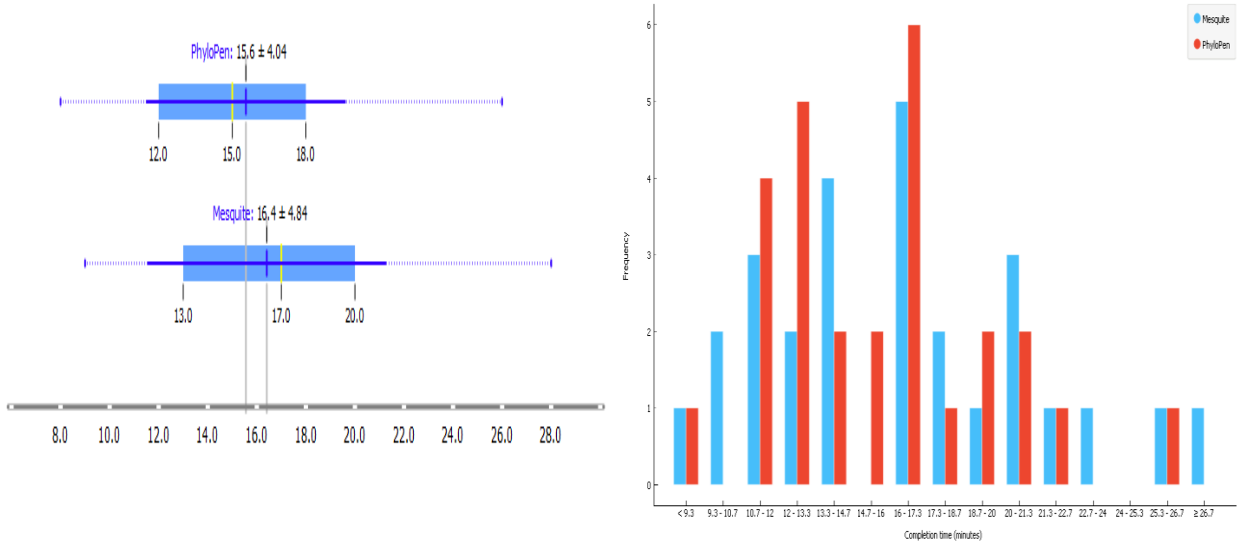


Figure 34: The difference in completion time between Mesquite and PhyloPen.

For all other dependent measures, the distributions were determined to be statistically significantly different by the Wilcoxon signed test. To determine further details, we use similar visuals to completion time. The difference in usefulness, satisfaction, ease of learning, and ease of use are illustrated in Figure 35, Figure 36, Figure 37, and Figure 38, respectively.

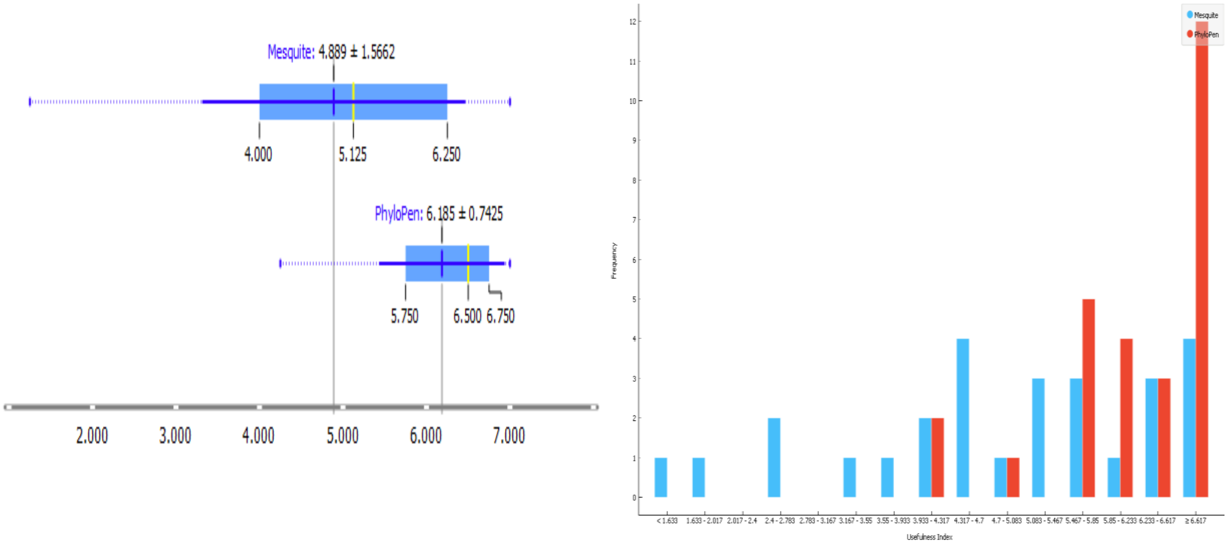


Figure 35: The difference in perceived usefulness between Mesquite and PhyloPen.

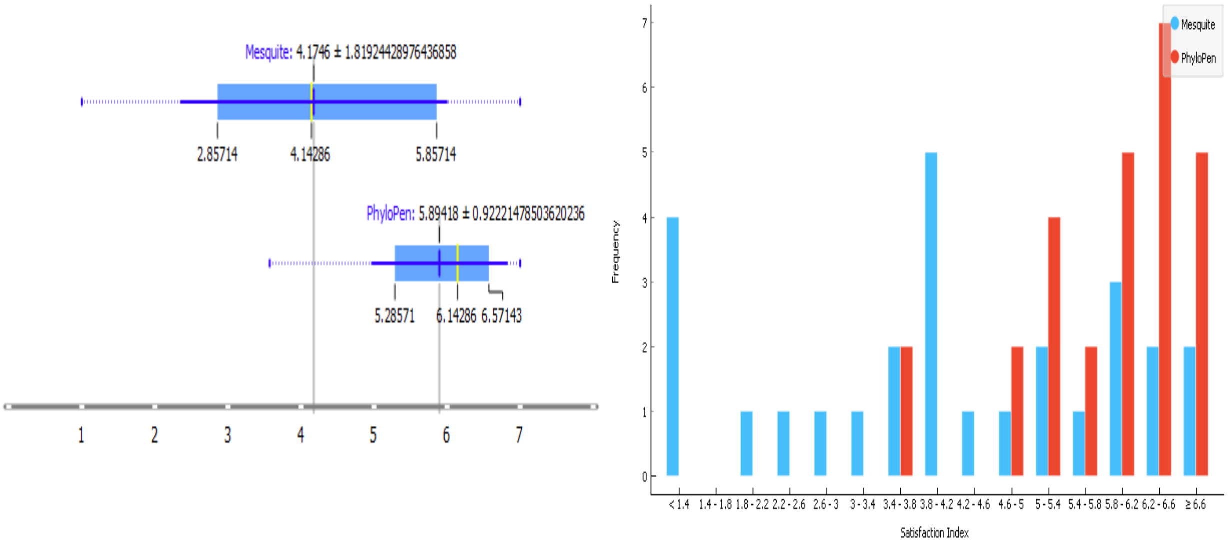


Figure 36: The difference in satisfaction between Mesquite and PhyloPen.

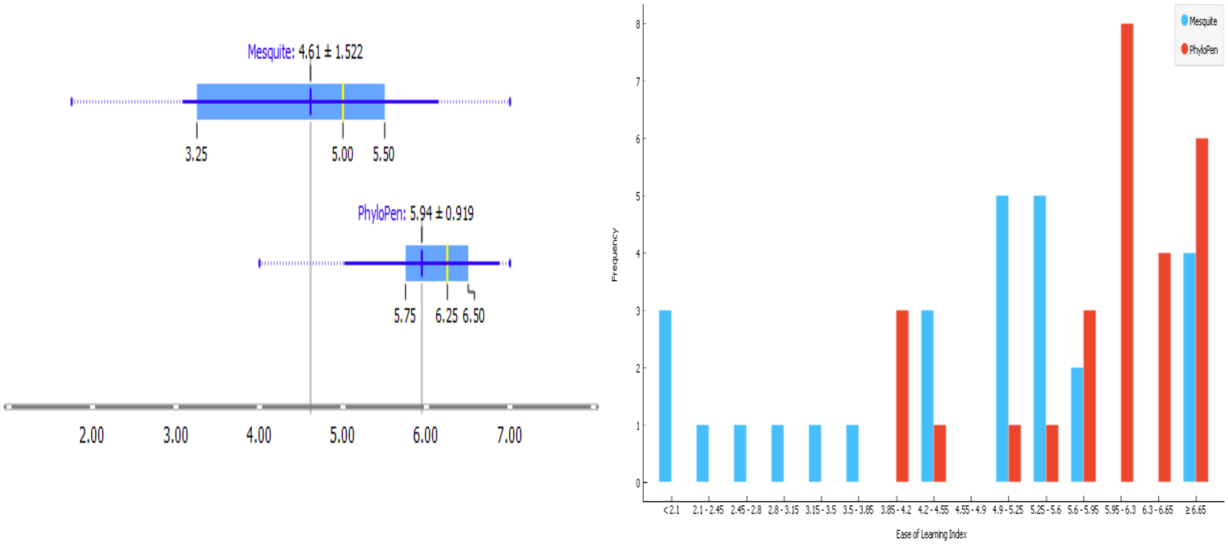


Figure 37: The difference in perceived ease of learning between Mesquite and PhyloPen.

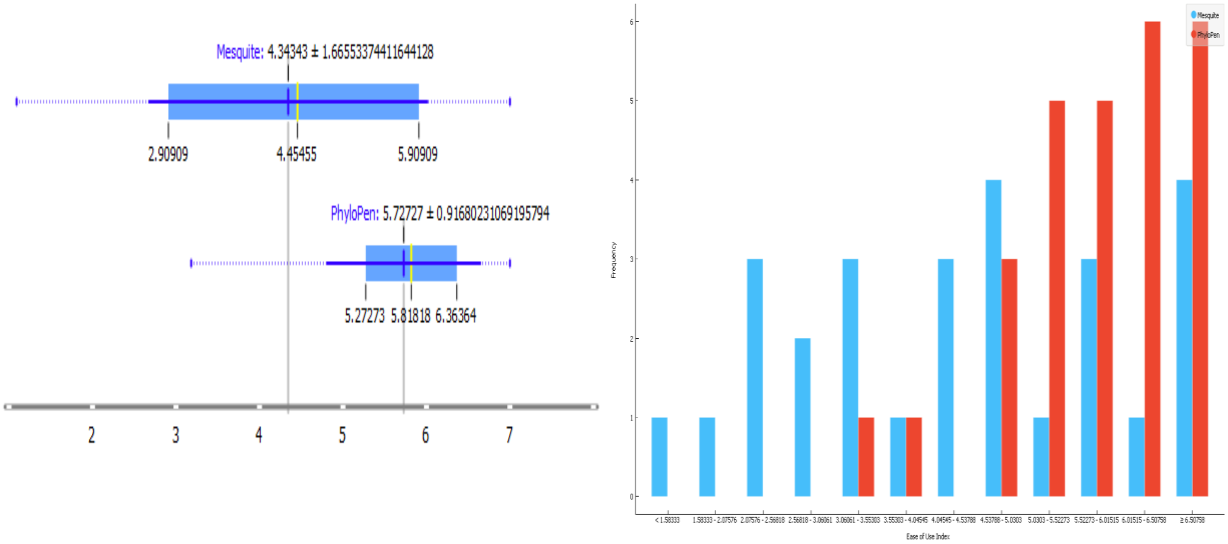


Figure 38: The difference in perceived ease of use between Mesquite and PhyloPen.

In each case, PhyloPen shows significant advantage over Mesquite. The mean and distribution consistently weigh higher on positive responses for PhyloPen with more variation in responses for Mesquite. It is largely the same for cognitive load, but since lower scores are better, the visuals are flipped in Figure 39. These results support H2, H3, H5, and H6 and reject H4.

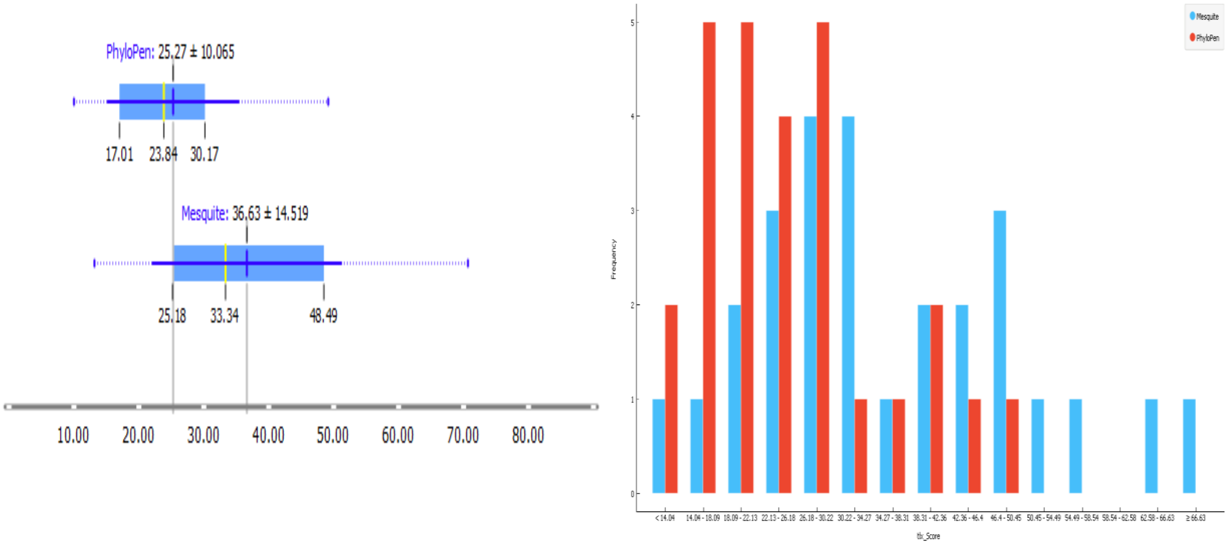


Figure 39: The difference in cognitive load between Mesquite and PhyloPen.

Additionally, we had measured for task completion by recording how many people decided to give up on an assigned task. Of all the sessions, only 4 people gave up on a task, one time each, with two quitting on exercise task 2-1 and two on exercise task 2-2. While all these instances occurred when using Mesquite, there were not enough occurrences to prove anything about H7. It is also important to note when observing these results that most participants found difficulty in these two tasks, making incorrect guesses frequently, whereas it was rare for them to make mistakes on the other exercise tasks once trained to perform them. When conversing with them during the training phase, the participants frequently expressed confusion on the structure of the tree. We intentionally left out family names in the phylogenetic tree structure, leaving it to be implied that they were somewhere in between. Additionally, the name labels they saw above the leaves could have been a subfamily name instead of a family. While we tried to verbally stress this point, it is likely that most participants (who were non-expert level students) were too new to tree thinking to understand this concept outright. If we presented them with a simpler and consistently labeled structure, it is possible they would have shown different results.

While we were able to test our hypotheses using the overall results, more observations can be made when considering the order in which the treatments were given. We calculated the mean for each dependent measure for each treatment when it was given to the participant first or second as shown in Table 8. If we observe the numbers, we find that, when PhyloPen is used first, Mesquite is judged much more harshly than if Mesquite were first. Additionally, between the two treatments being used first, PhyloPen scores better on average in each case (excluding completion time). Finally, the greater variance in Mesquite’s scores is consistent in all cases.

Table 8: The mean (M) and standard deviation (SD) for each dependent measure for each treatment and then when considering the order that each treatment was given to the participant.

	PhyloPen		Mesquite		PhyloPen First		Mesquite First		PhyloPen Second		Mesquite Second	
	M	SD	M	SD	M	SD	M	SD	M	SD	M	SD
Completion time (min)	15.5	4.12	16.4	4.93	16.8	4.39	18.3	4.71	14.4	3.59	14.1	4.32
Usefulness	6.19	0.81	4.89	1.60	6.08	0.81	5.57	1.33	6.29	0.72	4.03	1.52
Satisfaction	5.89	0.94	4.17	1.85	5.77	0.85	4.83	1.71	6.01	1.03	3.36	1.76
Ease of Learning	5.94	0.94	4.61	1.55	5.83	1.06	5.12	1.26	6.05	0.83	3.98	1.69
Ease of Use	5.72	0.93	4.34	1.70	5.66	0.82	4.80	1.62	5.79	1.05	3.77	1.68
Cognitive load	25.3	10.3	36.6	14.8	25.0	9.95	33.5	10.1	25.5	10.9	40.6	18.9

Finally, we can derive new observations from the final comparison portion of the post-surveys. In the final comparison, we asked them explicitly which software they preferred and which one they felt presented the data in a more meaningful way. We visualize the results in a pie chart in Figure 40. While PhyloPen is favored in both cases, there is some disparity between the two charts. The common stated reason from almost all participants for Mesquite being perceived as being superior in presenting the data is that Mesquite has a more compact and easy-to-read character matrix. PhyloPen, on the other hand, has a more spread out matrix that cannot necessarily be read all at once at a single glance and requires scrolling to view the rest of the data. As for why people generally preferred PhyloPen, we attempted to bin the responses and graph them in Figure 41. The most popular responses were that PhyloPen was easier or friendly to use, it had a better visual display that was easier to read, and it was also perceived as "modern" or "futuristic."

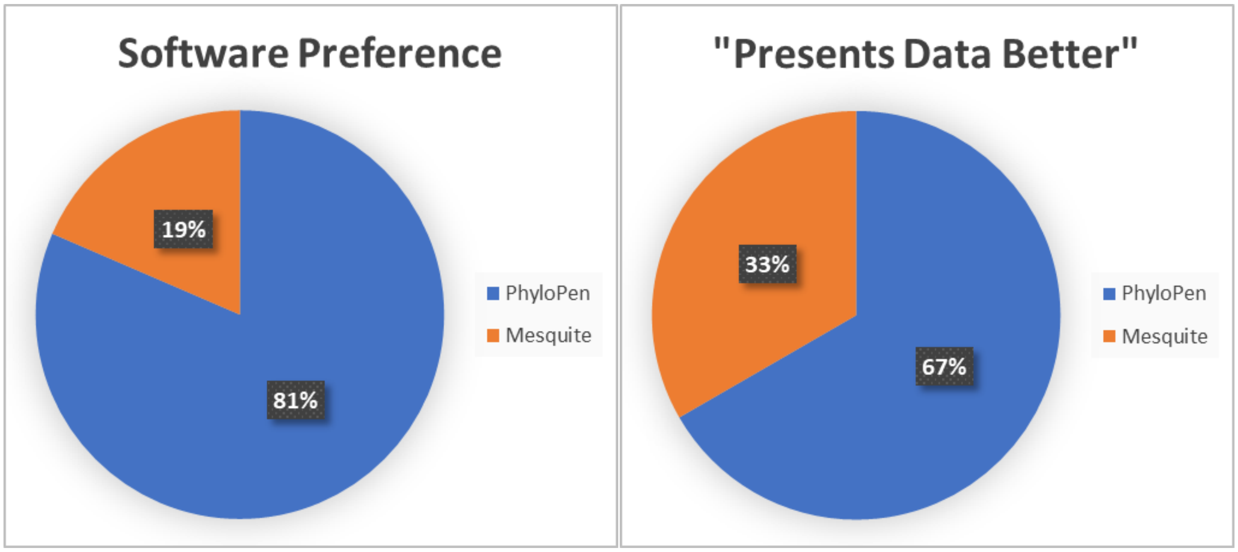


Figure 40: The final comparison results from the post-surveys of which software the participants preferred and which they thought presented the data in a more meaningful way.

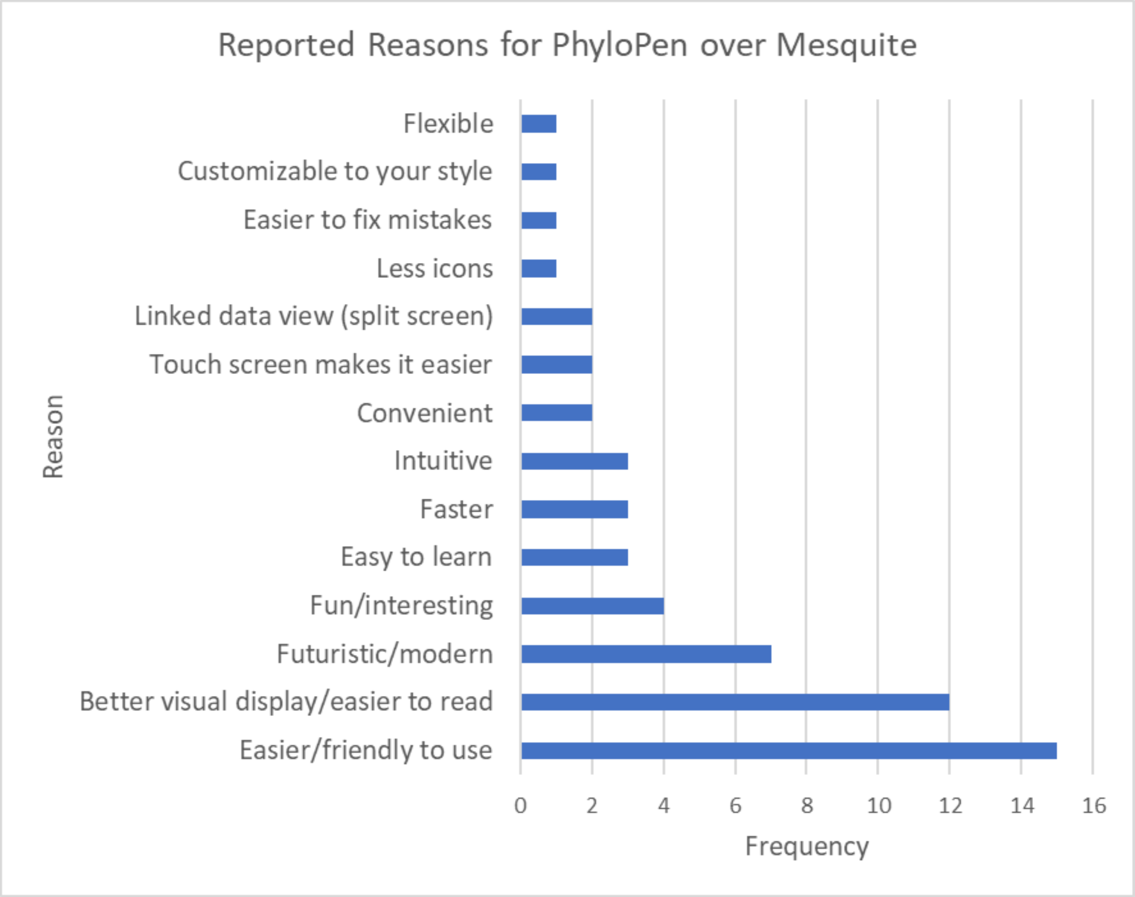


Figure 41: The reported reasons for why participants preferred PhyloPen over Mesquite.

Observing little difference in task success between Mesquite and PhyloPen, the overall results suggest PhyloPen creates a better sensemaking experience for users but do not suggest PhyloPen creates a more in-depth sensemaking experience than Mesquite, at least when faced with the limitations that were placed on it by the study, constraint that were enforced to draw a fair comparison between the two software systems in phylogenetic tree editing and viewing. For instance, it would not have been fair to compare when using larger trees as at some point they would become unworkable in Mesquite. We believe there is likely a better study format to evaluate increased sensemaking, and we leave that discussion for Chapter 6.

In this chapter, we presented a summative study for the effectiveness of PhyloPen in sensemaking of phylogenetic trees when compared to the gold standard of the field, Mesquite. We outlined the design of the study and presented the results of the study, showing that PhyloPen was rated higher in usefulness, satisfaction, ease of use, and ease of learning, lower in cognitive load, and relatively the same in completion time. We will discuss how we move forward from these results in future work as well as describing the research's broader impacts in the next chapter.

CHAPTER SIX: SUMMARY AND FUTURE DIRECTIONS

6.1 Summary of Results

The focus of this dissertation is on the development, analysis and refinement of multi-modal interaction paradigms associated with the visualization, management and manipulation of large, graph-connected datasets. While most work done here is in the specific context of phylogenetic trees, the techniques developed support sensemaking across a wide range of domains that work with large connected graphs. Our goal is, thus, the creation and investigation of natural and effective ways to improve user interactions with graphs that are large enough to not be viewable in detail as a whole. This complexity results in the need for intuitive means of focusing on subgraphs and/or of abstracting away details that are not germane to the user's current tasks. To this end, we experimented with different input modalities, designing and running a series of preliminary studies, ultimately focusing our attention on pen-and-touch. Through several iterations of feedback and revision provided with the help of biology experts and students, we were able to identify tree-based interaction paradigms that worked well and to improve on others that either were inefficient, cumbersome or, in some cases, even confusing, This led to our developing a pen-and-touch phylogenetic tree browsing and editing application called PhyloPen. This application expands on the capabilities of existing software with visualization techniques such as overview+detail, linked data views, and new interaction and manipulation techniques using pen-and-touch.

The primary research presented is a within-subject study that compares PhyloPen to Mesquite, the current gold standard for biologists who are visualizing and manipulating phylogenetic trees. Based on several metrics, our studies showed that the PhyloPen interaction paradigm was significantly better in terms of usefulness, satisfaction, ease-of-learning, ease-of-use, and cognitive load. Moreover, task completion was not adversely affected, nor was it improved by the use of PhyloPen.

6.2 Future Research Directions for PhyloPen

One of our original goals for this research was to develop tools that could be used to improve the teaching of phylogenetic concepts, leading to better tree-thinking by biology students, including those in high school. In fact, in an early study design we planned to bring the PhyloPen system into an AP Biology class. Unfortunately, access to longitudinal data that we felt was essential to our study was not available and the selected teachers were a bit uncomfortable with the potential loss of preparation time. Despite our disappointment in not be able to get that study off the ground during the dissertation, we feel it is a natural next step, especially with the system being improved and battle-hardened over the last several years.

While it was inconclusive whether PhyloPen increased the depth of sensemaking of phylogenetic trees in a short period of time, it could be tested for use over a longer time period. The comparison between PhyloPen and Mesquite may not be the most appropriate to test such hypotheses since Mesquite was an application specifically made for experts in the field and is generally never used as a teaching aid to help students learn. However, students were our primary targets in the final summative study. We had to specially set up the Mesquite environment to be user friendly to our audience. PhyloPen, on the other hand, is an application developed with students in mind as well as experts. A more concrete context to observe its use would thus likely be in the classroom. If a short test is too simple to show results, then we can observe how it affects student learning over a semester period, comparing grades with and without the treatment. The possible ways of setting up this type of experiment were discussed by past examples in Section 2.6. In addition, there is the possibility of bringing back the Tree-Thinking Challenge questions from one of our previous studies to test users' knowledge before and after.

One very natural use for PhyloPen in teaching and other communication events, e.g., seminars, conference talks and student projects, is as a presentation tool. In support of this, we have designed and partially implemented a set of tools that allows one to replay a tree editing session. In order to afford users this opportunity, we developed a recording/playback system and a set of features that allows one to take snapshots of an editing session. Each such snapshot is available as a starting point or a backup point. In particular, a user could make a presentation that involves visualizing multiple hypotheses, e.g., about parent-child relationships in some subtree. Each hypothesis would be recorded as a branch in a history tree that is organized as a cactus stack, a data structure first used in the Burroughs B5000 and B6000 mainframes to manage stack branching in concurrent systems (Organick, 1973). This allows the presenter to follow any path previously checkpointed or to go back to a checkpoint and start a new branch in the cactus stack.

Automatic tree creation is another feature greatly desired by the biology community and one that would be a great asset for learning. Pulling phylogeny data from potentially multiple databases, such as Open Tree of Life, Encyclopedia of Life, and the database maintained by OneZoom, we would like to create a usable tree structure and render it straight into PhyloPen simply by typing a few keywords of interest. For example, if a student were interested in learning about the relatedness of reptiles and birds, they could include them as keywords and the program would generate a tree displaying their relationship. This feature has been a work in progress already, but it could use more refinement. In particular, there are some interesting challenges here as the way data is stored in online databases may not always be appropriate for creating phylogenetic trees and must be preprocessed and cleaned before use for the information to make sense and be in the proper format for PhyloPen.

Finally, we would like to revisit the topic of exploring large phylogenetic trees. The ability to render and interact with large phylogenetic trees was one of original motivating purposes behind the Arbor Workflows project that started PhyloPen. From the author’s experience and personal perspective, PhyloPen has demonstrated the ability to render and handle larger trees better than any of the tools typically used by biology experts, but we believe there will eventually be a cut-off point to its ability to render. The ability to render much larger trees likely lies in techniques like those used by OneZoom, and that is to not display all information at once but only a general overview and then to display more information on a given point as the user approaches it. However, we would like to keep it in the traditional format that biology experts prefer rather than appealing to a more general audience as OneZoom does.

6.3 Broader Impacts

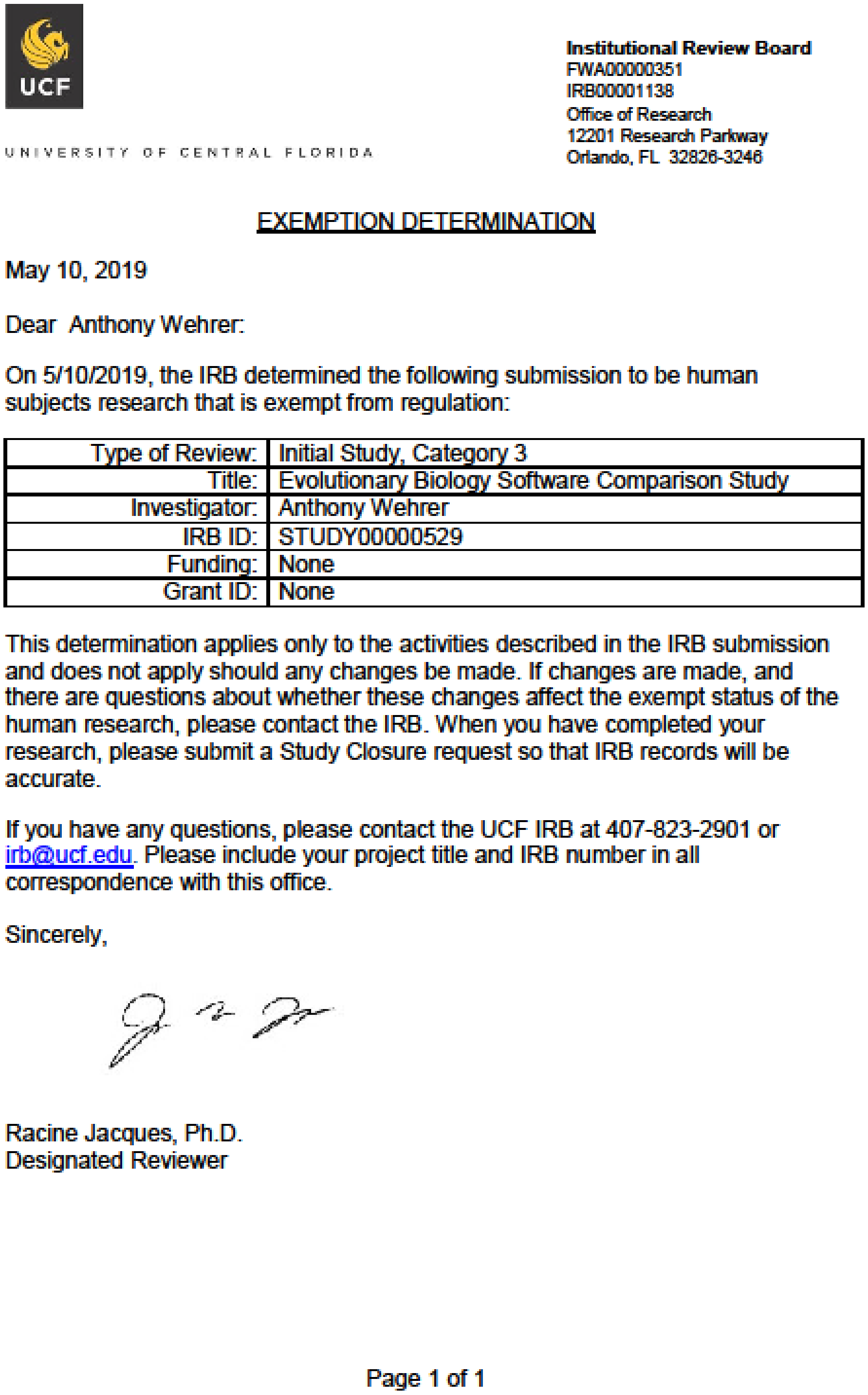
While the user interface we developed for PhyloPen is being used for phylogenetic trees, the model could also be extended to other fields of study that use trees or graphs, one of which can be illustrated with our experimental “NetPen” interface for social networks and power grids. As both application areas have graphs with attribute data, most of our gestures/features lend themselves well to these application areas, especially selection and ordering/coloring by attribute value, a feature we have designed and are now implementing in PhyloPen. A user could make notes on certain nodes with annotations in NetPen, and then translate and scale views for large grids of nodes. To keep track of their relative position in a large grid of nodes, the user would have to show all edges at times to find one to focus on instead of the default hidden setting. At this point, an overview+detail model could also be useful for users to avoid losing context. Expanding and collapsing nodes could work differently as well. Whether a node shows its edges could be considered an expanded or collapsed state. The user could use expand and collapse gestures to show edges leading out from only certain nodes at one time. Edges could be automatically colored based on dynamic or static attributes that may reflect the strength or perhaps critical nature of such connections.

Both social networks and electric power grids have edge attributes that might carry dynamic information. In the case of social networks, one unusual but potentially higher useful application relates to circles of trust, where children and their parents might establish the kinds of online interactions the children might have with trusted versus untrusted nodes. Direct connections might reflect trusted relations and indirect ones may even require mediation through trusted nodes. Changes to such a structure might lead to automatic notification to the parent. While discussions about the appropriateness of such restrictions is beyond the scope of our work, we feel that working with others, such as (Ghosh, et al., 2018) and (Wisniewski, Ghosh, Xu, Rosson, & Carroll, 2017), who are experts in this domain, could lead to very impactful broader impacts.

As regards electric power grids, we envision applications related to the monitoring of such systems. Today's Bulk Power industry manages power grids around the country, and these networks have many nodes (representing power stations) and edges (representing long-haul power transmission lines) with constantly changing state. In some cases, the operator needs a network overview, in other cases, the operator may want to only focus on a small section of the network at one time. A user monitoring the grid could use gestures to collapse parts of the graph to a single relevant area or have a command for the system to automatically expand critical areas based on current attributes about their states, their connections or even the states of nodes to which they are connected. Translation and zoom may also be used to view more details while the overview+detail miniature model prevents loss of context. The nodes could be colored by their power levels or some other attributes.

With so many potential application areas that use graphs, PhyloPen’s model could see applicability to a wide range of research problems in addition to its original focus on phylogenetic trees in biology.

APPENDIX A: IRB APPROVAL LETTER



UCF

UNIVERSITY OF CENTRAL FLORIDA

Institutional Review Board
FWA00000351
IRB00001138
Office of Research
12201 Research Parkway
Orlando, FL 32826-3246

EXEMPTION DETERMINATION

May 10, 2019

Dear Anthony Wehrer:

On 5/10/2019, the IRB determined the following submission to be human subjects research that is exempt from regulation:

Type of Review:	Initial Study, Category 3
Title:	Evolutionary Biology Software Comparison Study
Investigator:	Anthony Wehrer
IRB ID:	STUDY00000529
Funding:	None
Grant ID:	None

This determination applies only to the activities described in the IRB submission and does not apply should any changes be made. If changes are made, and there are questions about whether these changes affect the exempt status of the human research, please contact the IRB. When you have completed your research, please submit a Study Closure request so that IRB records will be accurate.

If you have any questions, please contact the UCF IRB at 407-823-2901 or irb@ucf.edu. Please include your project title and IRB number in all correspondence with this office.

Sincerely,

Racine Jacques, Ph.D.
Designated Reviewer

Page 1 of 1

APPENDIX B: VERBAL CONSENT FORM

EXPLANATION OF RESEARCH

Title of Project: Evolutionary Biology Software Comparison Study

Principal Investigator: Anthony Wehrer, Ph.D. Student, Department of Computer Science, University of Central Florida
Other Investigators: Arianna Davis, Undergraduate student, UCF

Faculty Supervisor: Charles E. Hughes, Professor, University of Central Florida

You are being invited to take part in a research study. Whether you take part is up to you. Those who choose to participate should have sufficient Biology background to have seen and have some basic understanding of Phylogenetic Trees.

The purpose of this study is to compare two software applications for visualizing phylogenetic trees in how effective they are in operability and conveying meaning to the user. We wish to identify what features work well between the two and apply these principles to future software tools to be made available to the evolutionary biology community.

The study session will take place in HEC 247C and will follow the outline listed:

- **Consent and Introduction**: Before the study session starts, we will give you this consent form and explain important information about the study before verbally agreeing to participate. You will then be asked to pick a number at random that will be used in the place of your name for information gathered during the study. If your name or contact information was gathered during the sign-up process, it will be destroyed after the study session has ended.
- **Pre-Survey**: We will provide you a short survey to gather demographic information and to understand what experience you have with evolutionary biology software prior to the study.
- **Software-A Training Exercise**: You will be asked to complete a series of tasks using one of the two software applications. You will receive an exercise sheet to gives direction in completing the tasks and a separate sheet that explains the relevant commands available. Additionally, if you have trouble, you will be able to ask us for help. These tasks are only for practice to make sure you understand how to use the software, but please try to work through them as quickly as possible.
- **Software-A Evaluation Exercise**: You will be asked to complete another series of similar tasks using the same software, but we will ask you to attempt to complete them on your own. You have the option to give up on a certain task at any point if it takes too long or you are frustrated. During this time, we will also ask you to try to think aloud as you solve the problem. Your voice will be recorded during this time so that we may later analyze how users interpret the software. Remember it is alright to admit if you are confused by the software as this is important for later improvement.
- **Software-A Post-Survey**: You will be given a survey to complete asking about your experience using the software. For the first post-survey, leave the 3rd section of the survey blank and scroll down to submit. This is the comparison section and will not be relevant until the final survey. The survey will not be finished until you hit the Submit button.
- **Software-B Training Exercise**: The same as the Software-A training exercises but with the other software application.
- **Software-B Evaluation Exercise**: The same as the Software-A evaluation exercises but with the other software application.
- **Software-B Post-Survey**: You will be given a survey to complete asking about your experience using the software. This time, you will complete the 3rd section of the survey.

Upon completion of the study, you will receive a $20 Barnes and Nobles gift card for your time.

Depending on work speed, the study should be completed in 45 to 90 minutes. Please work as quickly as possible to achieve the assigned tasks.

You will be audio recorded during this study. If you do not want to be recorded, you will not be able to be in the study. Discuss this with the researcher or a research team member. If you are recorded, the recording will be kept in a locked, safe place. The recording will be erased or destroyed when we have finished analyzing them.

Your participation in this study is voluntary. You are free to withdraw your consent and discontinue participation in this study at any time without prejudice or penalty. Your decision to participate or not participate in this study will in no way affect your relationship with UCF, including continued enrollment, grades, employment or your relationship with the individuals who may have an interest in this study.

Any results we publish will be kept anonymous and information about your participation collected by us will be completely confidential. All personally identifiable information such as names and contact information will be destroyed and the unique ID you chose at the start of the study will be left at the information's identifier. All the recorded audio data will be encrypted and password protected on a shared drive and will be available to study investigators only.

You must be 18 years of age or older to take part in this research study.

Study contact for questions about the study or to report a problem: If you have questions, concerns, or complaints: *Anthony Wehrer, Ph.D. Student, Department of Computer Science, (407) 916-9597 or by email at awehrer@knights.ucf.edu or Dr. Charles E. Hughes, Faculty Supervisor, Department of Computer Science by email at ceh@cs.ucf.edu.*

IRB contact about your rights in this study or to report a complaint: If you have questions about your rights as a research participant, or have concerns about the conduct of this study, please contact Institutional Review Board (IRB), University of Central Florida, Office of Research, 12201 Research Parkway, Suite 501, Orlando, FL 32826-3246 or by telephone at (407) 823-2901, or email irb@ucf.edu.

REFERENCES

Anderson, D., Bailey, C., & Skubic, M. (2004). Hidden markov model symbol recognition for sketch-based interfaces. *AAAI Fall Symposium*, pp. 15-21.

Anthony, L., & Wobbrock, J. O. (2010). A lightweight multistroke recognizer for user interface prototypes. *Graphics Interface (GI '10)* (pp. 245-252). Toronto: Canadian Information Processing Society.

Baum, D. A., Smith, S. D., & Donovan, S. S. (2005). The tree-thinking challenge. *Science, 310*(5750), pp. 979-980. doi:10.1126/science.1117727

Block, F., Horn, M. S., Philips, B. C., Diamond, J., Evans, M. E., & Shen, C. (2012). The DeepTree exhibit: visualizing the tree of life to facilitate informal learning. *IEEE Transactions on Visualization and Computer Graphics*, *18*(12), pp. 2789-2798. doi:10.1109/TVCG.2012.272

Brandl, P., Forlines, C., Wigdor, D., Haller, M., & Shen, C. (2008). Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. *Working Conference on Advanced Visual Interfaces: AVI '08*, pp. 154-161. doi:10.1145/1385569.1385595

Bryan, C., Guterman, G., Ma, K.-L., Lewin, H., Larkin, D., Kim, J., . . . Farre, M. (2017, January). Synteny explorer: An interactive visualization application for teaching genome evolution. *IEEE Transactions on Visualization and Computer Graphics, 23*(1), pp. 711-720. doi:10.1109/TVCG.2016.2598789

Cao, X., & Balakrishnan, R. (2005). Evaluation of an on-line adaptive gesture interface with command prediction. *Proceedings of Graphics Interface 2005 (GI '05)*, (pp. 187-194).

Card, S. K., & Nation, D. (2002). Degree-of-interest trees: a component of an attention-reactive user interface. *Working Conference on Advanced Visual Interfaces (AVI '02)* (pp. 231-245). New York: ACM. doi:10.1145/1556262.1556300

Cockburn, A., Karlson, A., & Bederson, B. B. (2009, January). A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys (CSUR), 41*(1). doi:10.1145/1456650.1456652

Cranston, K., Harmon, L. J., O'Leary, M. A., & Lisle, C. (2014, June 19). Best practices for data sharing in phylogenetic research. *PLOS Currents: Tree of Life*. doi:10.1371/currents.tol.bf01eff4a6b60ca4825c69293dc59645

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In L. Saitta (Ed.), *Machine Learning: Proceedings of the Thirteenth International Conference (ICML '96)*, (pp. 148-156).

Ghosh, A. K., Badillo-Urquiola, K., Xu, H., Rosson, M. B., Carroll, J. M., & Wisniewski, P. (2018). A matter of control of safety? Examining parental use of technical monitoring apps on teens' mobile devices. *Proceedings of the 2018 ACM Conference on Human Factors in Computing Systems (CHI 2018)*, Paper 194, 14 pages. doi:10.1145/3173574.3173768

Gorgevik, D., Cakmakov, D., & Radevski, V. (2001). Handwritten digit recognition by combining support vector machines using rule-based reasoning. *Proceedings of the 23rd International Conference on Information Technology Interfaces*, (pp. 139-144). doi:10.1109/ITI.2001.938010

Graham, M., & Kennedy, J. (2001). Combining linking and focusing techniques for a multiple hierarchy visualisation. *Fifth International Conference on Information Visualisation (IV '01)* (pp. 425-432). Washington, DC: IEEE. doi:10.1109/IV.2001.942092

Harmon, L. J., Baumes, J., Hughes, C. E., Soberon, J., Specht, C. D., Turner, W., . . . Thacker, R. W. (2013, June 21). Arbor: comparative analysis workflows for the tree of life. *PLOS Currents: Tree of Life*. doi:10.1371/currents.tol.099161de5eabdee073fd3d21a44518dc

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLK (Task Load Index): results of empirical and theoretical research. *52*, pp. 139-183. doi:10.1016/S0166-4115(08)62386-9

Heer, J., & Shneiderman, B. (2012, April). Communications of the ACM. *Interactive dynamics for visual analysis, 55*(4), pp. 45-54. doi:10.1145/2133806.2133821

Holzinger, A., Kickmeier-Rust, M. D., Wassertheurer, S., & Hessinger, M. (2009, February). Learning performance with interactive simulations in medical education:Lessons learned from results of learning complex physiological models with theHAEMOdynamics SIMulator. *Computers & Education, 52*(2), pp. 292-301. doi:10.1016/j.compedu.2008.08.008

Horn, M. S., Tobiasz, M., & Shen, C. (2009). Visualizing biodiversity with voronoi treemaps. *Sixth Annual International Symposium on Voronoi Diagrams in Science and Engineering: ISVD '09* (pp. 265-270). Copenhagen, Denmark: IEEE. doi:10.1109/ISVD.2009.22

Isenberg, P., & Carpendale, S. (2007). Interactive tree comparison for co-located collaborative information visualization. *IEEE Transactions on Visualization and Computer Graphics, 13*(6), pp. 1232-1239. doi:10.1109/TVCG.2007.70568

Isenberg, P., & Isenberg, T. (2013). Visualization on interactive surfaces: a research overview. *I-COM, 12*(3), pp. 10-17. doi:10.1515/icom.2013.0020

Kara, L. B., & Stahovich, T. F. (2004). An image-based trainable symbol recognizer for interfaces. *AAAI Fall Symposium*, pp. 99-105.

Kristensson, P.-O. (2004). SHARK2: a large vocabulary shorthand writing system for pen-based computers. *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)* (pp. 43-52). New York: ACM. doi:10.1145/1029632.1029640

LaViola Jr., J. J. (2005). *Mathematical sketching: a new approach to creating and exploring dynamic illustrations.* (Doctoral dissertation), Brown University, Providence, Rhode Island.

Lund, A. M. (2001). Measuring usability with the USE questionnaire. *Usability Interface, 8*(2), pp. 3-6.

Maddison, W. P., & Maddison, D. R. (2019). *Mesquite: a modular system for evolutionary analysis*, Version 3.6. Retrieved from Mesquite Project: http://www.mesquiteproject.org

Munzner, T. (2008, November). Process and pitfalls in writing information visualization research papers. In A. Kerren, J. T. Stasko, J.-D. Fekete, & C. North (Eds.), *Information visualization: human-centered issues and perspectives* (pp. 134-153). Berlin, Germany: Springer. doi:10.1007/978-3-540-70956-5_6

Munzner, T., Guimbretiere, F., Tasiran, S., Zhang, L., & Zhou, Y. (2003, July). TreeJuxtaposer: scalable tree comparison using focus+context with guaranteed visibility. *ACM Transactions on Graphics, 22*(3), pp. 453-462. doi:10.1145/882262.882291

Myers, C. S., & Rabiner, L. R. (1981, September). A comparative study of several dynamic time-warping algorithms for connected-word recognition. *The Bell System Technical Journal, 60*(7), pp. 1389-1409. doi:10.1002/j.1538-7305.1981.tb00272.x

Notowidigdo, M., & Miller, R. (2004). Off-line sketch interpretation. *AAAI Fall Symposium*, pp. 120-126.

Organick, E. I. (1973). *Computer System Organization: The B5700/B6700 Series.* New York, NY: ACM Monograph Series, Academic Press.

Paradis, E. (2006). *Analysis of Phylogenetics and Evolution with R.* New York: Springer.

Pittman, J. A. (1991). Recognizing handwritten text. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)* (pp. 271-275). New York: ACM. doi:10.1145/108844.108914

Plaisant, C., Grosjean, J., & Bederson, B. (2002). SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation. *IEEE Symposium on*

Information Visualization (InfoVis'02) (pp. 57-64). Washington, DC: IEEE. doi:10.1109/INFVIS.2002.1173148

Plamondon, R., & Srihari, S. N. (2000, January). On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(1), pp. 63-84. doi:10.1109/34.824821

Rambaut, A. (2016, October 4). *FigTree (Version 1.4.3) [Software]*. Retrieved August 11, 2017, from http://tree.bio.ed.ac.uk/software/figtree/

Rosindell, J., & Harmon, L. J. (2012, October 16). OneZoom: A fractal explorer for the tree of life. *PLOS Biology, 10*(10). doi:10.1371/journal.pbio.1001406

Rubine, D. (1991). Specifying gestures by example. *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91)* (pp. 329-337). New York: ACM. doi:10.1145/122718.122753

Schneider, B., Strait, M., Muller, L., Elfenbein, S., Shaer, O., & Shen, C. (2012). Phylo-Genie: engaging students in collaborative 'tree-thinking' through tabletop techniques. *SIGCHI Conference on Human Factors in Computing Systems: CHI '12* (pp. 3071-3080). New York: ACM. doi:10.1145/2207676.2208720

Schweitzer, D., & Brown, W. (2007). Interactive visualization for the active learning classroom. *38th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '07)* (pp. 208-212). New York: ACM. doi:10.1145/1227310.1227384

Sezgin, T. M., & Davis, R. (2005). HMM-based efficient sketch recognition. *Proceedings of the 10th International Conference on Intelligent User Interfaces* (pp. 281-283). New York: ACM. doi:10.1145/1040830.1040899

Sharek, D. (2009). *NASA-TLX online tool (version 0.06) [Internet Application]*. (Research Triangle, NC) Retrieved from http://www.nasatlx.com

Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. *IEEE Symposium on Visual Languages (VL '96)* (pp. 336-343). Washington, DC: IEEE. doi:10.1109/VL.1996.545307

Shneiderman, B., & Plaisant, C. (2005). *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (4th ed.). Boston: Pearson Addison-Wesley.

Stasko, J., Choo, J., Han, Y., Hu, M., Pileggi, H., Sadana, R., & Stolper, C. D. (2013). CiteVis: exploring conference paper citation data visually. *IEEE Information Visualization Conference (InfoVis).* Atlanta, GA.

Stieff, M., & Wilensky, U. (2003, September). Connected chemistry - incorporting interactive simulations into the chemistry classroom. *Journal of Science Education and Technology, 12*(3), pp. 285-302. doi:10.1023/A:1025085023936

Tappert, C. C. (1982, November). Cursive script recognition by elastic matching. *IBM Journal of Research and Development, 26*(6), pp. 765-771. doi:10.1147/rd.266.0765

Teow, L.-N., & Loe, K.-F. (2002, November). Robust vision-based features and classification schemes for off-line handwritten digit recognition. *Pattern Recognition, 35*(11), pp. 2355-2364. doi:10.1016/S0031-3203(01)00228-X

Vatavu, R.-D., Anthony, L., & Wobbrock, J. O. (2012). Gestures as point clouds: a $P recognizer for user interface prototypes. *14th ACM International Conference on Multimodal Interaction* (pp. 273-280). New York: ACM. doi:10.1145/2388676.2388732

Walny, J., Lee, B., Johns, P., Riche, N. H., & Carpendale, S. (2012). Understanding pen and touch interaction for data exploration on interactive whiteboards. *IEEE Transactions on Visualization and Computer Graphics, 18*(12), pp. 2779-2788. doi:10.1109/TVCG.2012.275

Wehrer, A., Yee, A., Lisle, C., & Hughes, C. E. (2015, November 23). PhyloPen: phylogenetic tree browsing using a pen and touch interface. *PLOS Currents: Tree of Life*. doi:10.1371/currents.tol.d6d666469fc1942c665cb895b2305167

Wisniewski, P., Ghosh, A. K., Xu, H., Rosson, M. B., & Carroll, J. M. (2017). Parental control vs. teen self-tegulation: is there a middle ground for mobile online safety? *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17),* 51-69. doi:10.1145/2998181.2998352

Wobbrock, J. O., Wilson, A. D., & Li, Y. (2007). Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. *20th Annual ACM Symposium on User Interface Software and Technology* (pp. 159-168). New York: ACM. doi:10.1145/1294211.1294238

Wolin, A., Eoff, B., & Hammond, T. (2008). ShortStraw: a simple and effective corner finder for polylines. In C. Alvarado, & M.-P. Cani (Ed.), *Fifth EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (SMB '08)* (pp. 33-40). Aire-la-Ville, Switzerland: EUROGRAPHICS Association. doi:10.2312/SBM/SBM08/033-040

Wu, H.-K., Krajcik, J. S., & Soloway, E. (2001, September). Promoting understanding of chemical representations: students' use of a visualization tool in the classroom. *Journal of Research in Science Teaching, 38*(7), pp. 821-842. doi:10.1002/tea.1033

Xiong, Y., & LaViola Jr., J. J. (2009). Revisiting ShortStraw - improving corner finding in sketch-based interfaces. In C. Grimm, & J. J. LaViola Jr. (Ed.), *EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling (SBIM '09)* (pp. 101-108). New York: ACM. doi:10.1145/1572741.1572759

Yu, G., Smith, D. K., Zhu, H., Guan, Y., & Lam, T. T.-Y. (2017, January). ggtree: an r package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods in Ecology and Evolution, 8*(1), pp. 28-36. doi:10.1111/2041-210X.12628

Zhao, B., Liu, Y., & Xia, S.-W. (2000). Support vector machine and its application in handwritten numeral recognition. *Proceedings of the 15th International Conference on Pattern Recognition*, *2*, pp. 720-723. doi:10.1109/ICPR.2000.906176