ECES-450/650 Final Project Report

Analysis and Visualization of Pathogen Sequence Data

Submitted to,

Dr. Gail Rosen

Project by,

Toshika Fegade, MS in Electrical Engineering

Trung Hoang, BS/MS in Electrical Engineering

Hoa Nguyen, BS/MS in Computer Engineering

Steven Khoa, BS/MS in Computer Engineering

Joshua Boisvert, BS in Electrical Engineering

Date: June 12th, 2020

ABSTRACT

As the novel coronavirus commonly referred to as COVID-19 has spread throughout the world, there has been a push within the scientific community to understand how rapidly the virus can spread, how the spread can be modeled, which groups are most at risk, and which subtypes of the virus are most deadly. Given the dire nature of a global pandemic, substantial effort has been taken by microbiologists to both analyze and visualize all the data related to COVID-19 that is being generated daily. While much focus in the research literature is on understanding COVID-19 itself, some experts are attempting to come up with better ways to model and predict viral spread and visualize viral genome sequence data. This paper is concerned primarily with the latter. Using the EpiCoV data from the GISAID initiative's website, phylogenetic trees were constructed using all of the COVID-19 sequence data from human hosts that also contained the sex, age, and native country of the host. Contained herein is an attempt at using the Robinson-Foulds distance metric to compare trees generated using the FastTree and RAxML algorithms (using the online CIPRES tool), as well as the use of the CLC Main Workbench program to integrate each of the aforementioned metadata categories into the phylogenetic visualization as annotations. The goals for this project were originally to use these tools to determine which subtypes of COVID-19 were more prevalent among particular groups of people or geographic regions. The idea is to determine whether or not a significant correlation exists between any of the clades in the trees (subtype of COVID-19) and the different metadata categories. For clade-level comparison of different hosts, we propose the use of the R packages 'treeio' and 'ggtree'.

LITERATURE REVIEW

It should be stated explicitly that the entirety of the Literature review section is not our work. The first paper that we considered was a paper on the soft Robinson Foulds distance to use as a metric to compare phylogenetic trees. For over two decades, research was done in the area of phylogenetic networks, trees, and clusters to understand how these things relate to one another. Two problems mainly occurred - tree containment problem (whether a phylogenetic tree is displayed in phylogenetic network or not) and cluster containment (whether cluster is represented at a node in phylogenetic network). There was an algorithm generated for the cluster containment problem developed and implemented in C. This was further used for the fast computation of the soft Robinson – Foulds distance between phylogenetic networks. In phylogenetic trees, the taxa below a node form a unique subset of taxa called its cluster. Measuring the dissimilarity between phylogenetic networks is important for assessing a networks reconstruction method. One of the metric functions proposed was the Robinson-Foulds distance.

Algorithms from the paper can be generalized to determine whether a cluster C is in a phylogenetic network N or not. It selects a non-trivial exposed component M of N. If M is visible, we will find a negative answer to the problem by working on M, or we obtain an instance of the problem that is simpler than the input instance (C,N) in linear time proportional to the size of M. In the latter, we reduce the original instance of the CCP to a simpler instance. If M is not visible, there is then a reticulation node which has a unique leaf child and does not have all parents in M. In this case, two phylogenetic networks N_1 and N_2 are derived from N, which contain fewer nodes than N. The algorithm is then called on both instances (C, N_1) and (C, N_2) recursively. Though algorithm seems simple, it has significantly less time complexity then for a binary input network.

The CCP algorithm was used to compute the Soft Robinson Foulds distance between two arbitrary networks on the same taxa set X. First, a k-cluster is defined with k taxa. Enumeration of all possible clusters over taxa X is done by generating all k-clusters of X for each k ranging from 1 to X-1. The time complexity of this SRF distance algorithm is $O(2^{|L(N)|}\ T(N))$, where T(N) is the time complexity of the CCP algorithm. To understand the performance of CCP algorithm, an example of five groups of networks is explained in paper with 10 leaves and all possible (2^{10}-2) clusters. Each group has 20 networks and network in k^{th} group had 5(1+k) nodes for each k from

1 to 5. This gave wall clock time 15 minutes and 15 seconds on average, so the program took about one centisecond for each pair.

The conclusion of comparing the SRF distance with the RF distance gave a few points – a) There are at least as many soft clusters as clusters in a network. Therefore, as expected, the SRF distance has larger range than the RF distance. b) The RF distance seems to have a normal distribution with a small mean and variance. c) The distribution of the SRF distances seems not to be normal. It is skewed towards small distances (especially for networks with more leaves) and a small fraction of network pairs had much larger SRF distances than the average SRF distance. Therefore, it is indicated that SRF distance is fine metric for networks and hence more suitable than RD distance for measuring dissimilarity of networks.

Another paper that we considered was to understand different visualization methods out there. We did not use this method, but it came under literature review since this gave us good options mentioned in the paper such as iTOL, ETE, etc. This paper talks about a web-based technology developed in order to allow interactive displays of complex data sets. Such phylogeny.IO web server is able to import, annotate and share interactive phylogenetic trees. This serves as an advantage since usually sharing phylogenetic trees is a task and has a lot of issues such as – splitting tree across various journals, additional data layers or time calibrations. This makes difficult for users to grasp. Developed method allows easy and rapid sharing of figures in blogs, lecture notes, press releases, etc.

The main advantage of this method is ease of use. There is no programming experience required, website maintenance or account registration is not necessary as well. The format is inspired by FigTree software package. The goal was to develop web server using JavaScript libraries, specifically D3.js for phylogenetic tree display. As a result, the application works with modern browsers, including mobile devices, and allows interactive phylogenetic trees to be easily embedded in static web pages for sharing. Visualization is performed client-side by web browser, decreasing computational load and complexity on the server hosting the tress. All the data used to render and annotate trees are accessible to user, so the process is reproducible and transparent.

This platform currently supports Newick and NEXUS files produced by BEAST, PhyloXML and custom JSON format that is used for exporting annotated trees. Once the tree has been annotated, it can be shared as an HTML iframe document and embedded into any static web

page. There are other features included such as adding confidence intervals to nodes and changing size, color and appearance of features using extended data in tree file. The disadvantage of using JavaScript for visualizing large data sets is images must be rendered in web browsers locally. This limits size of tree that can be used. The visualization is limited to 4000 leaf nodes total and only 1000 leaves will be displayed simultaneously. This was the main reason why we did not opt for this method. But this method gave us insight in what features we can look for and what other software are out there for such visualization technique.

Phylogenetic trees are used to depict evolutionary relationships. This is needed to compare multiple large trees inferred from the same set of taxonomic data. This reflects uncertainty in the tree interference or any genuine discordance from the analyzed phylogenetic tree. Existing visualization tools are not well suited. Some analyze only specific organisms, such as fungi or bacteria only. *Phylo.io* is a web application that can visualize and compare phylogenetic data. This is good because of challenges from other tools that result from comparing trees for a few taxonomies. *Phylo.io* can highlight similarities and differences of two trees, automatically identifies the best matching root and leaf order, has high scalability for large trees, is able to be used on multiple platforms because it is implemented on HTML5, has the ability to share and store visualizations, has high usability, and is free to use. This is useful for analyzing the COVID-19 virus data from GISAID because it is a free tool that can be shared easily, especially when most people are in quarantine. Work on analyzing branches of phylogenetic data can be done in your own home. Two other popular tools for tree visualization are FigTree and EvolView. Both tools can display and manipulate tree visualizations, but they can only analyze a single tree at a time, and they cannot compare different topologies. Other tools can display multiple trees, such as SplitsTree and it represents trees as a network, but it is difficult to pinpoint the specific changes between two trees, making it not a good tool for COVID-19 where we have multiple trees and associated metadata that we want to associate with them.

The *Phylo.io* tool is accessible and usable on all modern web browsers. This was good for our group because we were all working from home because of the quarantine. It was also able to share tree visualizations using GitHub. The tool also makes it easier to read by having some nodes collapse and having others that are not visible to keep an organized and readable tree. The tool can improve the legibility of large trees by being able to collapse the nodes so that an overview of the

tree remains visible at any given time. This is needed for the metadata attributes for the COVID-19 hosts we were trying to analyze to make more sense out of the visualization. Multiple tools were used for analyzing the data because of the amount of sequences and metadata information that the COVID-19 sequences had on GISAID. Comparing two trees need to match the leaves in one tree or find the best corresponding node in the other tree. This is to find topological similarities between the data you are analyzing. The reference generated two trees using different methods, one was with PhyML and the other one was RAxML. These trees contained proteins and were able to show that the trees were different only because of the different rooting and subtree ordering. (https://arxiv.org/ftp/arxiv/papers/1602/1602.04258.pdf)

Another way of shaping trees to be more readable and to see the characterization of the shapes of the rooted branching trees are to make metrics for phylogenetic tree shapes. The metric distinguishes trees from random models known to produce different tree shapes. Metrics are an appealing way to compare sets of objects or traits. Defining a metric defines a space for the set of objects or traits. The labeling scheme for metrics can characterize tree shapes. Tree shapes are similar if they share many subtrees with the same label.

The reference analyzed an influenzas sequence to reflect different epidemiology of tropical and the seasonal flu. This can be done for the COVID-19 strain too because of it being an influenza, but more branches needs to be analyzed from COVID-19 to be able to find the problems of the virus to help reduce the spread and to find where it came from. DNA sequencing has been becoming more available and the cost of doing it has been declining. Diversity, variation, and evolutionary traits of organisms is more widely available then before. Metrics are down on the space of rooted unlabeled shapes to compare tree shapes from different models of evolution or different sets of data. Metrics needed to be used on the large dataset of COVID-19 because of the new virus. A tree shape is a tree without additional information of tip labels and the branch lengths. Rooted trees have one node specified at the root. The reference simulated data on the HA protein sequences from the human influenza A (H3N2) strain was analyzed and aligned with mafft. This is different from the Bayesian interference of many trees in a set of tips. The reference incorporated the tree size, branch lengths, and other properties. The developed metrics in the reference was used to analyzed unlabeled tree shapes and compared simulated and data-driven trees. (https://academic.oup.com/sysbio/article/67/1/113/3788885)

The last kind of visualization we want to mention is Nextstrain which is a real-time tracking of pathogen evolution. Nextstrain consist of a database of viral genomes, a bioinformatics pipeline for phylodynamic analysis, and an interactive visualization platform. All these are available on their website https://nextstrain.org/. This tool allows user to observe the real-time view of the evolution and spread of a range of viral pathogens of high public health importance including Covid-19. Moreover, the tool support visualization for different geographic location, serology, and host species.

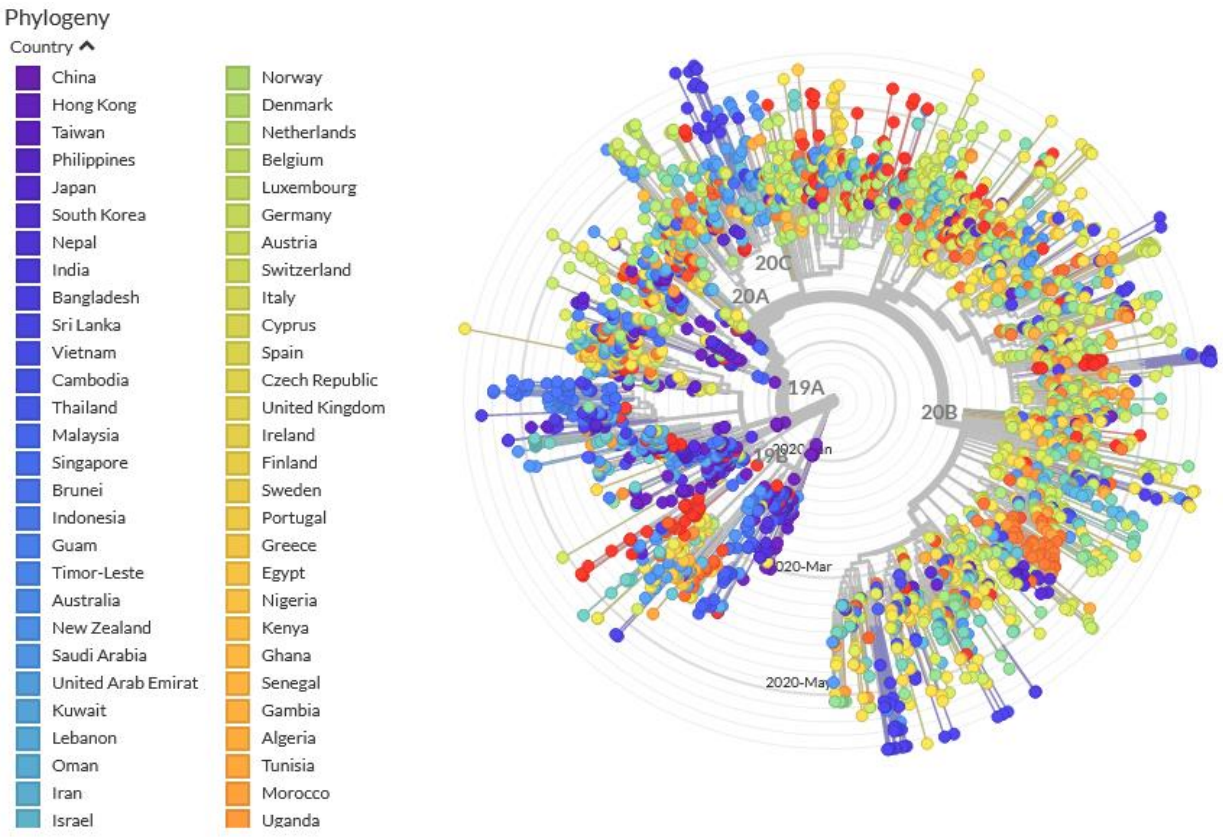

Figure 1. An Example of visualization available on Nextstrain.org. Source: https://nextstrain.org/ncov/global

This visualization is particularly helpful to observe how different strains of virus are found in different countries because it shows countries as a gradient of colors. Countries that are geographically closer to one another are closer in terms of color which helps to identify which virus is prominent in which countries or regions. Moreover, the tool also has a slider where user can change the range of date of information they want to see.

MATERIALS AND METHODS

In this project, we used two datasets. One is from GISAID which is a global science initiative and primary source of genomic data. On GISAID, we used the EpiCoV dataset which contains COVID-19 virus genome sequences. As of June 12th, there are more than 46,000 sequences on GISAID EpiCoV. However, we started our project back in May when there were only about 16,000 sequences, so these entries are the only ones that will be discussed in this project. The second dataset we used is the metadata published by Nextstrain.org on their GitHub repository. As of June 12th, Nextstrain has moved this metadata to GISAID website and both metadata and sequence data can be accessed through GISAID front-end. The first dataset which contains the virus sequences was used to construct the trees. The second dataset was used to label the created trees. Out of the 16,000 sequences, we only needed sequences from "human" hosts with known "age", "sex", and "country". Therefore, we filtered both datasets based on these criteria. This step was done with Python3 and the code is available on the project repository (https://github.com/trung-hn/covid-19). From the 16,000 sequences, there are ~8,300 sequences that have the metadata we are interested in.

We then used CIPRES Science Gateway to perform the next step which was to align the sequences and create the trees. CIPRES stands for Cyberinfrastructure for Phylogenetic Research which is a project that helps scientists interact with the XSEDE supercomputer. On CIPRES, we used MAFFT to align the sequences. MAFFT works by converting the amino acids into vectors of volume and polarity. The frequency of the amino acid substitutions depends on the difference of the volume and polarity of their physio-chemical properties. FFT (Fast-Fourier Transform) goes from the time domain to the frequency domain. For alignments sequences, the FFT shifts the sequence by 'k' until the peaks correspond to two homologous blocks. Although MAFFT is not as fast as MUSCLE, they have comparable accuracy and MAFFT supports up to 30,000 sequences which is ideal for our project. This process took roughly 6-7 hours for a 251 MBs fasta file, which grows to 326 MBs post-alignment. After obtaining the aligned sequences, we created trees using two different techniques; RAxML and FastTree. Both RAxML and FastTree can be performed on CIPRES directly, however we only used CIPRES to perform RAxML. For FastTree, we decided to utilize the multithreaded version on a 32-core machine to speed up the process. RAxML on CIPRES took ~2 days to complete while FastTree took ~7 hours on our workstation.

After preprocessing the data and making a proper table, we needed to prep the data for visualization. The next goal for the project was to align the datapoints with their respective metadata and color label the trees for each metadata category. A phylogenetic tree is a diagram that represents evolutionary relationships among organisms. Phylogenetic trees are hypotheses, not definitive facts. The pattern of branching in a phylogenetic tree reflects how species or other groups evolved from a series of common ancestors. A cladogram is a type of phylogenetic tree that only shows tree topology — the shape indicating relatedness. Cladograms are concerned with the way organisms are related to common ancestors through shared characteristics. We worked on labeling trees according to metadata and visualization of the tree. We used multiple approaches for visualization such as – iTOL, ViPR (Virus Pathogen Resource), Qaigen CLC Workbench, ETE Toolkit, etc.

iTOL visualizes the tree with a bootstrap range and threshold. There was an option to delete branches having bootstrap values less than the threshold. When the option of distinguishing branches by color was selected, it only gave us 3 options – minimum (red), midpoint (yellow), and maximum (green). The entire tree was represented by using only 3 colors according to the bootstrap values, which did not give us enough information to create subtrees or leaves and branches to work with. The metadata can be visualized in 4 different ways: (1) Using Symbols: select the shape type, minimum and maximum, size and color. (2) Text: values will be displayed as text labels on the branches. Font size, numeric style and position of the label can be adjusted. (3) Color: tree branches will be colored according to their metadata values and the color gradient defined in the control box. For non-numeric metadata sources, colors will be assigned automatically. (4) Width: tree branch widths will be set according to their metadata values. Minimum/maximum widths selected will be used as the widths for the branches with minimum/maximum metadata value. Dataset option can be used to add Metadata, but free trial supports only 5 entries for tree. So we could not move forward with this option.

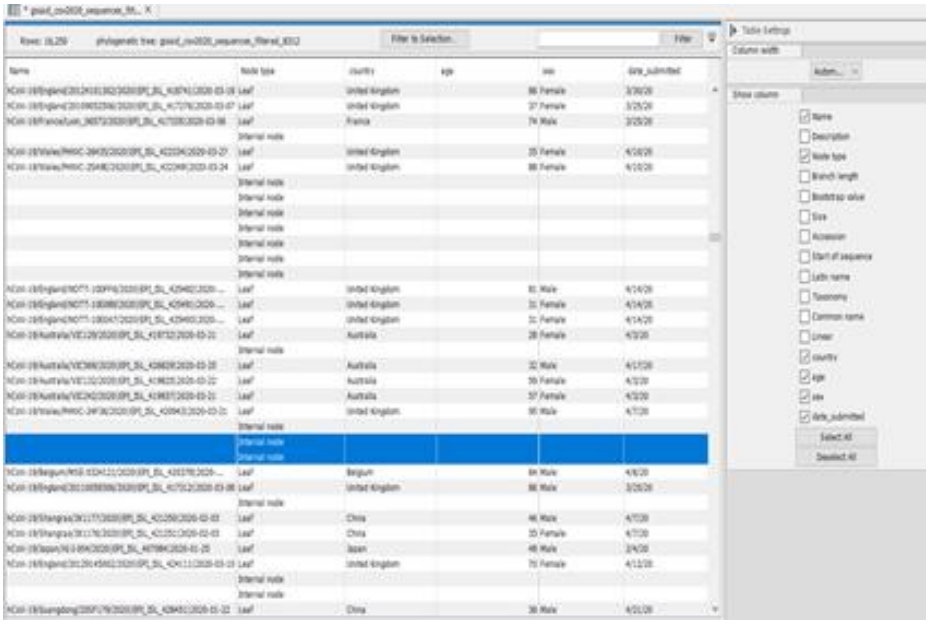

Figure 2 Data table in CLC workbench

ViPR (Virus Pathogen Resource) contains some online features such as Meta-driven Comparative Analysis Tool, Visualize Aligned Sequences and Generate Phylogenetic tree. We were focusing on working with Metadata-driven Comparative Analysis Tool (meta-CATS) tool. The meta-CATS tool provides the capability to perform customized comparative genomics analyses with minimal manual manipulation. It is possible to perform a statistical analysis on sequences assigned to up to 10 different groups to determine which residues significantly correlate with one or more metadata fields. The meta-CATS tool looks for positions that significantly differ between user-defined groups of sequences. But unfortunately, the data upload demanded a lot of time and this tool did not respond as we expected. We needed to find another way for visualization.

The final method that we considered for labeling and visualization step is the QIAGEN software CLC Main Workbench. We used a free trial license which delivered promising results. We were able to align data points from the tree along with the metadata. In the context of the CLC Main Workbench, this usually means information about the samples. For example, a set of reads could come from a specimen at a time point with some given characteristics. The specimen, time, and characteristics would be the metadata for that set of reads. In our case, it was age, date, and country. Metadata association - The data elements' associated metadata rows can be listed by selecting the metadata rows of interest. Inheritance of metadata associations requires that a single association can be unambiguously identified for an output when a tool is run. If an output is derived

from two or more inputs with different metadata associations, then no association will be inherited.

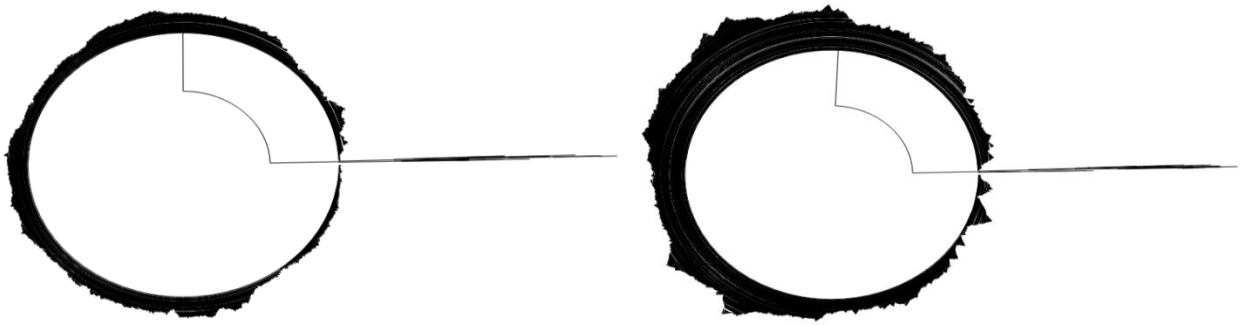

Figure 3. Fasttree (left) RAxML (right)

There are two ways in which associations can be done – manually or automatically. We opted for the automatic way for convenience sake. It is possible to visualize tree branches according to various metadata categories which makes it easier for data processing.

Our initial goal was to find useful metrics after generating a single tree using the FastTree method, but after closer inspection our tree size was too large to get any meaningful information from it. We then decided to create a second tree using a different method (RAxML) and find metrics to compare the differences between the two different way of creating the trees. The metric we decided on to compare the two trees was the Robinson-Foulds distrance metric. After generating the two trees with the same data using FastTree and RAmML we can visually see the difference between the two trees generated with two different methods, but we cannot draw any useful conclusion from it so we decided to look into metrics that can be used to compare trees. One metric we found in researching was the Robinson Foulds distance metric.

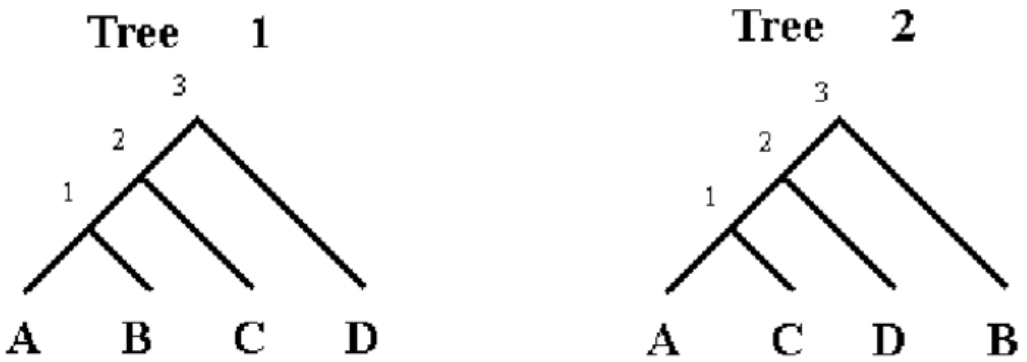

Figure 4. Trees 1 and 2 have 2 clades that do not exist in the other one (excluding the node 3). Hence the Robinson-Foulds distance is 2 + 2 = 4 in this example

The Robinson-Foulds distance between two trees T1 and T2 with n tips is defined by the following equation:

$$d\text{-}T_1, T_2 = i(T_1) + i(T_2) - 2v_s(T_1, T_2)$$

Where $i(T_1)$ denotes the number of internal edges and $v_s(T_1, T_2)$ denotes the number of internal splits shared by the two trees. The normalized Robinson-Foulds distance is derived by dividing $d(T_1, T_2)$ by the maximal possible distance $i(T_1) + i(T_2)$. If both trees are unrooted and binary this value is 2n-6.

```
(base) hoa@hoa-ubuntu:~/Desktop/ECES650Project$ python compareTree.py
Traceback (most recent call last):
  File "compareTree.py", line 17, in <module>
    rf, max_rf, common_leaves, parts_t1, parts_t2 = t1.robinson_foulds(t2, unroo
ted_trees=True)
ValueError: too many values to unpack
```

Figure 5. Out of date error given by ETEToolkit

To calculate and find the Robinson-Foulds distance we used a built-in method in the ETE Toolkit; the robinson_foulds() method. This method was designed to return the robinson-foulds distance between the two trees, along with any partition that exists in one tree but not the other. When running the given demo python script, we ran into an issue that the toolkit was out of date and cannot handle the tree size we given as input and return and error. This was unfortunate as this was a test run which we planned to use to determine if we could find the dissimilarity between any two trees. In the results section below, the prospect of instead doing a clade level analysis of the entire tree that we generated is considered, as we weren't able to collect any results that we could use to confirm or deny our original hypothesis.

RESULTS

As mentioned in the Materials and Methods section, multiple tools for phylogenetic tree visualization were used. QAIGEN CLC Workbench gave promising results in terms of metadata association with the data points. Metadata categories considered were age, country, and gender. Metadata age along with data points associated is shown below, along with a zoomed in version for better visualization:

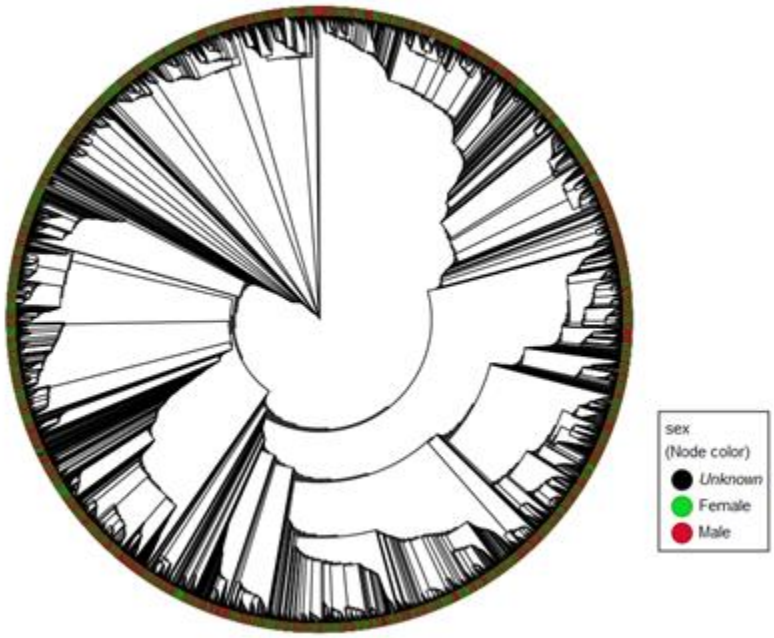

Figure 6

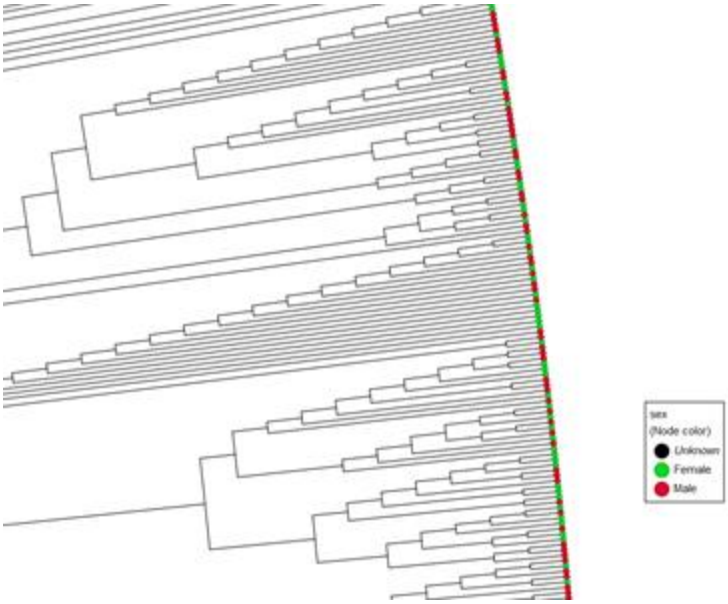

Figure 7

A phylogenetic tree with two layers of metadata (age and country) is shown below in figure 8:

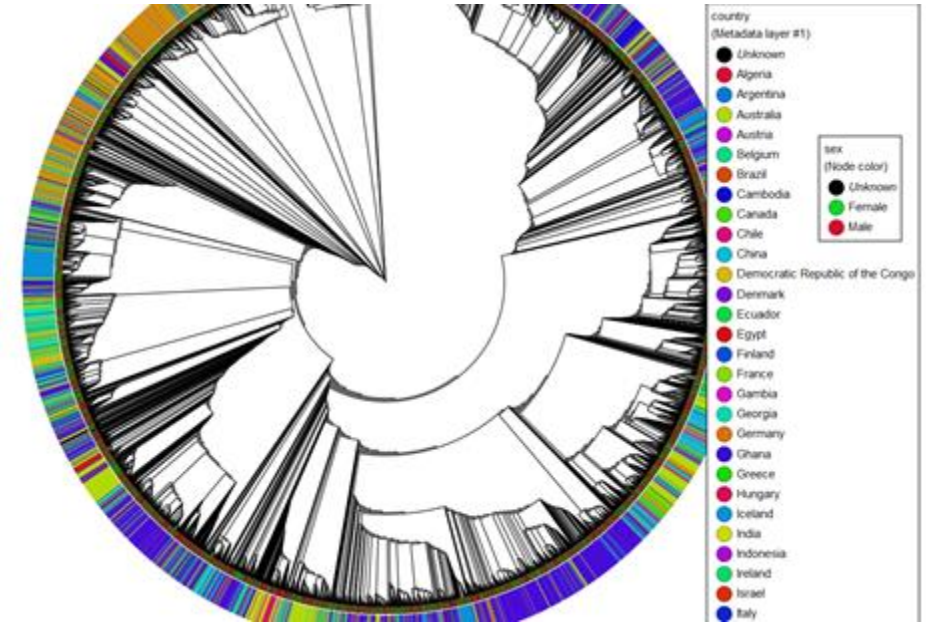

Figure 8

The zoomed in version gives a better visualization for nodes. If you click on the node, all the details pop up and it is easier since we do not have to go back to the table to look for the node properties.

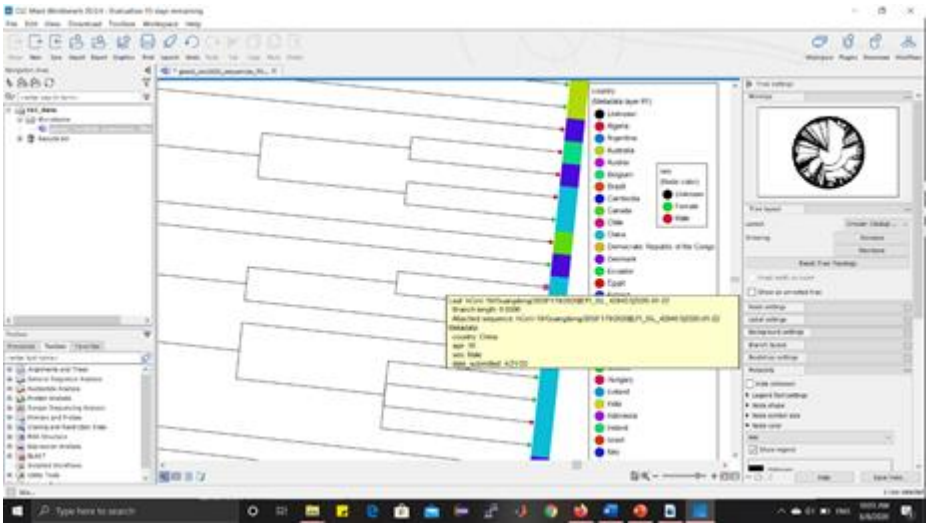

Figure 9

It was very interesting to work on this software since they had advanced options for metadata association and phylogenetic tree options. There were multiple layout options such as

rooted, unrooted, cladogram, and phylogram. Trees shown above follow rooted circular cladogram. There are multiple options for different visualizations such as node color, label color, and branch color. You can select colors individually for your preferences. This makes it convenient to work on a tree with lesser points but for higher numbers of points, analysis becomes difficult.

Since no metric was successfully implemented (hence no associated metric-based results), here we will briefly discuss an algorithm (as a high-level overview) for comparing the metadata for each of the clades of a given tree using the 'treeio' and 'ggtree' R packages (reasons for working with these packages are discussed in more detail in the discussion section of this report):

1. Read any Newick formatted trees into R using the command; read.newick(filename, node.label = "label", ...) where the 'label' arguments are optional (default labeling involves simply labeling all the nodes numerically)
2. Create a merged-tree object to incorporate the metadata in with the tree (accepts .csv format), see chapter 2 section 2.2 of the documentation for code samples. This step would effectively create a map from the nodes of the Newick file to the data, in our case these identifiers would be the ISL identifiers for each sequence from GISAID
3. Align the merged graph object (node labels+ tree structure) with the metadata by using a similar approach as the one found in Chapter 7.2-7.3
4. Using the methods found in Chapters 6.1-6.4, R can create an iterator (foreach loop) over a single clade and all of its branches. Simple conditional logic can be used to determine which data category each branch belongs to and count them up. This can be repeated over any of the clades in the tree
5. Using code from section 5.1-5.2, the results found for any of the clades can be added to the visualization of the tree as an annotation

Since these R packages are so powerful and customizable, there is a trade-off between the flexibility offered by these packages and the complexity of implementing them effectively. These packages are open source and are the recommendation from the team for this kind of work, as this software can be used to show many interesting qualities of a phylogenetic tree with associated metadata, and scripts can be written (not necessarily so easily) to iterate over any/all clades of a phylogenetic tree.

DISCUSSION

CLC Genomics Workbench is a powerful tool that features algorithms widely used by scientific leaders in the industry and academia. CLC Workbench is available on MacOS, Linux, and Windows. We utilized CLC Workbench in our project because it has an easy-to-use file structure and rich metadata labeling features. With this tool, we can import the tree in Newick format and import metadata in csv format and connect them easily. In CLC Workbench, when a data element is associated with a metadata row, the outputs of the analysis involving that data often inherit the metadata association automatically. In our case, this was an advantage because the tool automatically connected nodes/leaves of trees to its associated metadata and from there we displayed the metadata in different ways, such as different sizes of nodes/branch/text, different colors of nodes/branch/text, or different shapes of nodes/branch/text. CLC Workbench allowed us to quickly visualize the metadata using different criteria in different ways. An example of tree labeling is shown in the figure below.

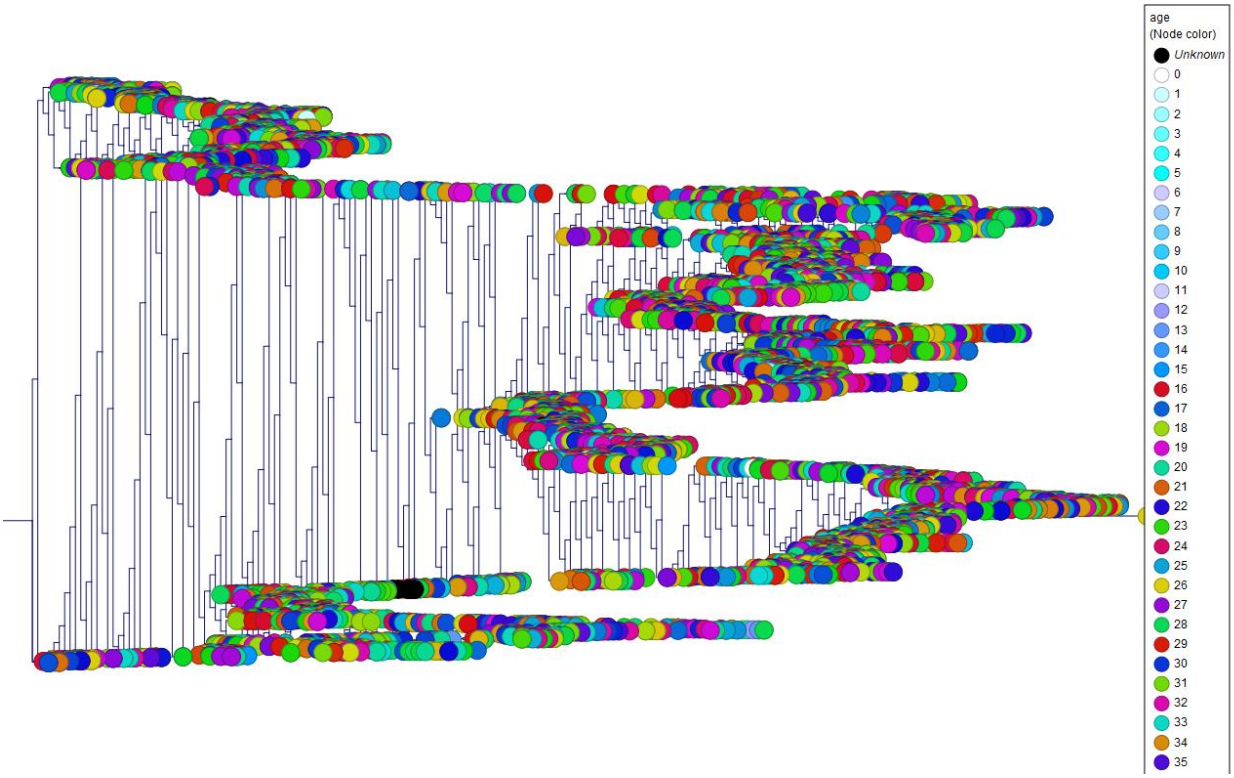

Figure 10. Tree is labeled by age with different colors using CLC Workbench

As you can see from figure 10, although the tool is useful for labeling tree when there are only a few categories, we have hundreds of categories in this project e.g. many different countries

and many different ages. This is where the tool is not as helpful as we expected. Based on the official manual of CLC Workbench, currently, there is no option to assign a gradient of colors to different values of a metadata. Although we can assign each value to a specific color, this is a very tedious task for more than 100 different values in ages and more than 50 different values in countries. In fact, if you look at the legend of the figure above, you can see that ages from 0 to 16 has a gradient and we did this manually. In order to proper visualize the metadata in meaningful way, we need a more refined tool where we can manipulate data on a bigger scale like a gradient of colors or groups of countries. Even so, CLC Workbench is useful for metadata with a few unique values such as gender as shown in figure 11.

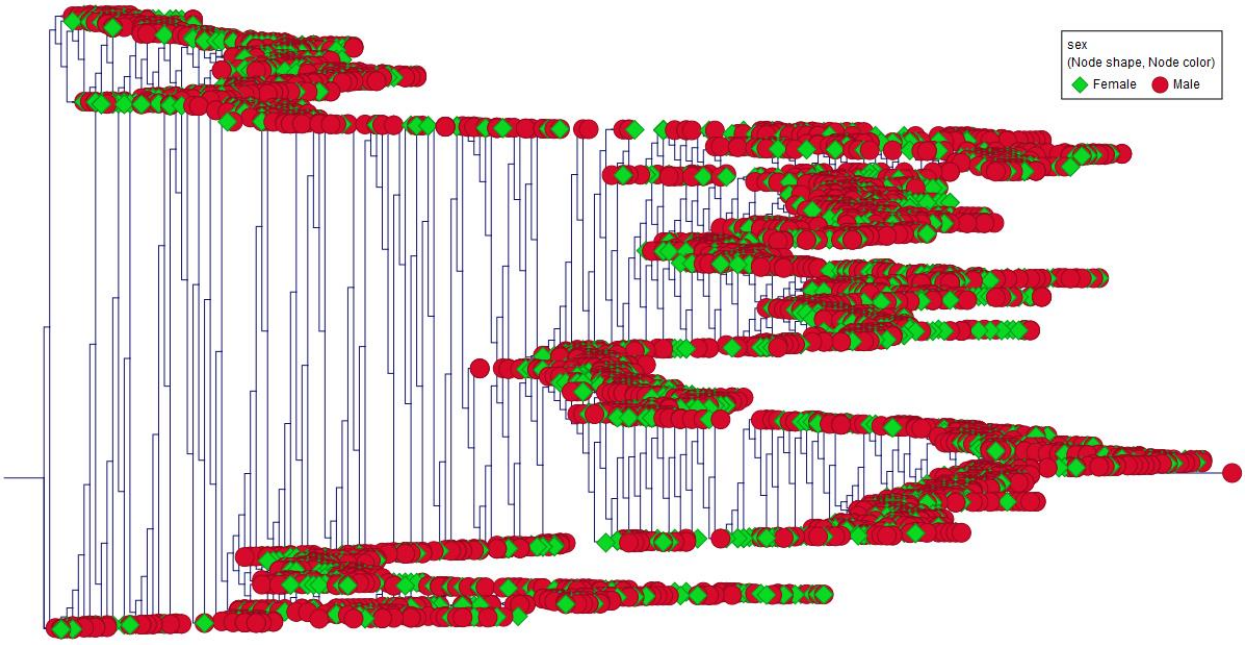

Figure 11. Tree is labeled by gender with different colors and shapes using CLC Workbench

The ‘treeio’ and ‘ggtree’ packages in R allow efficient input and output of Newick formatted files and flexible and programmatic control over data annotation of trees, integrating metadata associated with a tree alongside the base of it, real-time plotting and bar-chart visualization of metadata, clade-level selection, visual scaling of clades, clade-based tree traversal and rotation, and dynamic collapse/expansion of trees. Unfortunately, the discovery of this tool simply came too late in the project timeline for it to be feasible to analyze our tree/metadata using

it. While these R packages are ideal for the analysis that we would like to perform, the team did not have enough time to write the kind of R script that would be needed to disprove our hypothesis.

The application of this tool to this project would've come in the form of a tree traversal algorithm that could travel along each branch of the tree, keep track of how many of the hosts on the current clade belong to each metadata category, and then annotate each of the highest-level clades with a brief summary of the metadata on all hosts beneath that clave (X males, Y females, etc.). Also ideal for our purposes would be aligning our metadata with the leaves of the tree, as shown below in Figure 12.

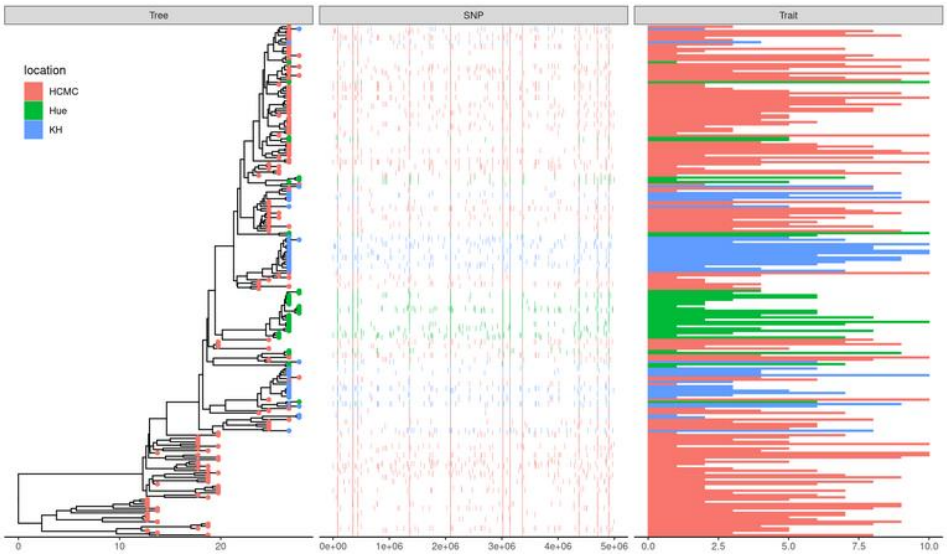

Figure 12. An example (from the ggtree documentation) of a Bar Chart of How Data Categories are Distributed Throughout the Tree

For our trees, it may instead to be more useful to show an age-range histogram for several claves. The code for accomplishing this task is relatively complex, but can be found in chapter 7 of the documentation which is included as a reference below.

It's worth mentioning that the results included above do not include the (successful) implementation of any metric associated with the data. It should be made clear that while it was not obvious from any of the tree visualizations if there exists a correlation between COVID-19 subtypes and the metadata, we cannot make a claim as to whether or not such a correlation exists. Instead, we suggest the use of the previously mentioned R packages and the approaches outlined in the Results section of this report to properly evaluate our hypothesis. However, based on the

visual overview of the data from CLC workbench, we don't believe any strong correlation exists between the metadata categories we looked at and the topology of the different COVID-19 sequences based on their alignment.

REFERENCES

ArXiv.org E-Print Archive. Accessed June 13, 2020. https://arxiv.org/ftp/arxiv/papers/1602/1602.04258.pdf.

CIPRES. "About." Portal | CIPRES. Accessed June 13, 2020. https://www.phylo.org/index.php/portal/about.

CLC Genomics Workbench. "USER MANUAL." Accessed June 13, 2020. https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/User_Manual.pdf.

"Data Integration, Manipulation and Visualization of Phylogenetic Trees." Site Not Found · GitHub Pages. Last modified June 2, 2020. https://yulab-smu.github.io/treedata-book/index.html.

GISAID. GISAID - Initiative. Accessed June 13, 2020. https://www.epicov.org/epi3/frontend#4d35cb.

Guang Chuang, Yu. "Data Integration, Manipulation and Visualization of Phylogenetic Trees." Site Not Found · GitHub Pages. Last modified June 2, 2020. https://yulab-smu.github.io/treedata-book/index.html.

Hadfield, James, Colin Megill, Sidney M. Bell, John Huddleston, Barney Potter, Charlton Callender, Pavel Sagulenko, Trevor Bedford, and Richard A. Neher. "Nextstrain: real-time tracking of pathogen evolution." *Bioinformatics* 34, no. 23 (2018), 4121-4123.

Hoang, Trung. "Trung-hn/covid-19." GitHub. Last modified 12, 2020. https://github.com/trung-hn/covid-19.

Lu, Bingxin, Louxin Zhang, and Hon W. Leong. "A program to compute the soft Robinson–Foulds distance between phylogenetic networks." *BMC Genomics* 18, no. S2 (2017).

"Nextstrain/ncov." GitHub. Last modified 13, 2020. https://github.com/nextstrain/ncov.

Robinson, Oscar. ArXiv.org E-Print Archive. Accessed June 13, 2020. https://arxiv.org/ftp/arxiv/papers/1602/1602.04258.pdf.