Genetic Algorithm for Phylogenetic Time Trees

Simon Austin Lee* Daniele Venturi
Jack Baskin School of Engineering
University of California, Santa Cruz
siaulee@ucsc.edu, venturi@ucsc.edu

https://github.com/Simonlee711/Research/tree/master/GeneticAlgorithm/2021

Abstract

The SARS-CoV-2 (COVID-19) outbreak has seen many mutations since it was first reported on December 31, 2019 in Wuhan China. Our primary research objective is to provide a representation of one of its variants, the deltacoronavirus and put it in a gene map known as *Phylogenetic Time Trees*. Our focus is to use a novel method in building distance matrices from a metaheuristic called the *Genetic algorithm*. This algorithm is inspired by Charles Darwin's Natural Selection and it is very strong in navigating through massive search spaces. It finds optimal solutions to combinatronic problems that take a lifetime to compute. Using genome data offered by the National Center of Bioinformatics Information (https://www.ncbi.nlm.nih.gov/datasets/coronavirus/genomes/), we build a time tree of this Coronavirus variant.

1 Introduction

SARS-CoV-2 (COVID-19) has changed the landscape of the world today. On March 11,2020 it was declared a global pandemic, threatening many lives globally. Scientists have been racing to not only develop a vaccine, but also construct accurate evolutionary diagrams. Therefore the field of phylogenetics has been crucial in tracking the relationships back to its origin. Though its origin still remains ambiguous, scientists have been able to track some of their closely-related evolutions using a series of alignment techniques to find small differences between genomes. Of the many representations, *phylogenetic time trees* are the focus of this research. Being seen as an evolutionary time map, we can see a small depiction of its evolution relative to time. Mathematically we can also visualize them as a dynamical system.

Our species of interest is not the coronavirus, but a subcategory of it. The Deltacoronavirus variant was first reported in India in late 2020 and has been spreading widely across the USA. Unlike the original virus, the delta variant is proven to be more contagious. Due to a slight change in its structure as seen in **Figure 1**, this has changed how the virus interacts with our cells. Once this variant comes into contact with a host receptor, it is actually better in terms of efficiency and at infecting them. For this reason the transmission of this particular virus has been widespread. And because of its current relevancy, we have chosen to use this species to model for our representation of the phylogenetic time tree.

2 Genetic Algorithm

The Genetic Algorithm (GA), is a metaheuristic algorithm inspired by Charles Darwin's theory of Natural Selection. This machine learning algorithm uses three components (Fitness, Single-Point Crossovers, Mutations) and is primarily used to solve optimization problems. With this method being stochastic meaning, it relies on randomness, our solutions do not have a definitive outcomes and are solely approximations. Our reasoning to choose this specific methodology is due to its performance in navigating massive search spaces. With our intentions to work with 112 different delta variant genomes, we need to consider the overwhelming number of combinations there

*Supported in part by the UCSC Applied Mathematics Department

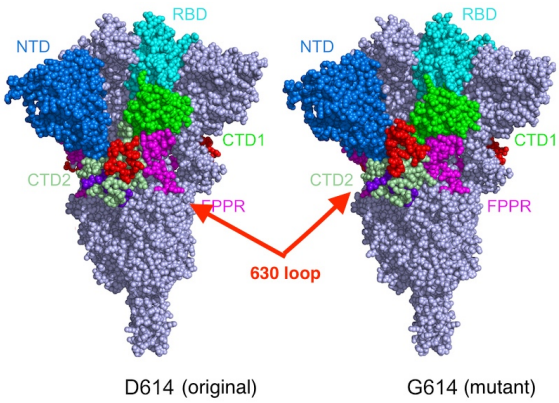

Figure 1: Pictured: SARS-CoV-2 Spike protein structure, Delta Variant Spike protein structure [1]

are to construct our tree. Governed by the equation below we plug in $n = 112$ to resemble the amount of genomes used in this research:

$$\frac{(2(n)-3)!}{2^{n-3}(n-3)!} \tag{1}$$

$$\frac{(221)!}{2^{109}(109)!} \tag{2}$$

$$\frac{5.047331342371357620848E423}{6.490371073168534535663E32(1.443859583202493582205E176)} = 5.386012149439842115461E214 \tag{3}$$

As seen in Equations 1,2,3 the following computation shows the amount of different combinations in which we could construct this tree. And as expected it is exceedingly overwhelming. Therefore machine learning algorithms like the GA have features like backwards propogation, and gradient descent that allow these solutions to learn the context of their data and provide better solutions as a result. With computation allowing us to be able to solve these massive combinatorics problems, this field is at the brinks of its capabilities. Regardless, the GA is a novel approach in solving these problems and we wish to use its output data to construct our trees. Below in **Figure 2** is a flowchart of how the processes within the algorithm all work together. These components will be explained in greater detail in the coming sections.

2.1 Fitness

"Survival of the Fittest" is a commonly used phrase in evolutionary biology to express how only the selected best solutions/species will make it to the next generation. As a result, we need to implement a similar implementation into our algorithm which evaluates how well a solution is. With our objective to line up these genomes based on how similar they are from one to another, our Genetic algorithm is assessing fitness based on how the sequence is lined up. Overtime these solutions should slowly begin to rearrange themselves in an order that embodies that of the distance matrix we intend to use. Visually we can see this improvement as seen in **Figure 3**.

2.2 Single-Point Crossovers

The second part of this algorithm involves how these solutions actually gain a higher fitness score. The only way to continue to have improved results is to perform a reproduction method known as the *single-point crossover*. In evolutionary biology, gene conversion can be *allelic*, meaning that one allele of the same gene replaces another allele from another gene. Similarly in this function, we perform a swap of a random number of genomes $1 \leq x \leq$ (number of genomes - 1). This will then swap a portion of the two randomly selected solutions and create a new solution whose order is closer to that of an optimal solution. These two newly generated solutions begin what would be the next generation. We repeat this process as long as there are remaining genomes from the past generation.

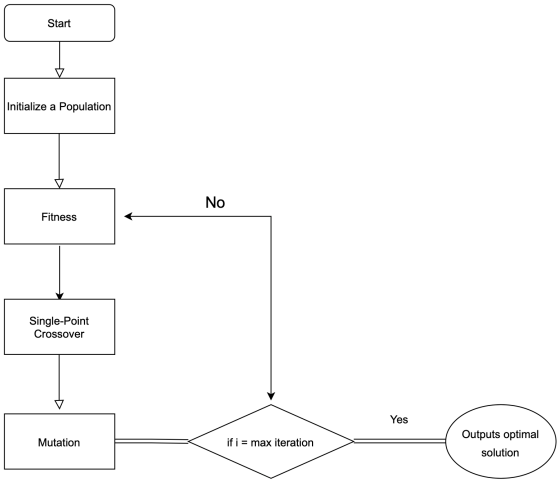

Figure 2: Flowchart of the Genetic Algorithm

However, this algorithm unfortunately is not perfect because our single-point crossover and selection functions are governed by randomness. The reason randomness is an issue is because there is no way to guarantee that we won't destroy our best solutions produced from each preceding generation. To resolve this issue, we must introduce a process called *elitism* [2]. Elitism can be described as, the most fit handful of solutions are guaranteed a place in the next generation - without undergoing mutation. So, in order to preserve the most fit solutions, we must select the top 10 genomes whose contents will be copied into is successive generation.

2.3 Mutation

The last component as alluded to in the previous section is mutation. Mutations in evolutionary biology are simply a change in a sequence of an organism's DNA. However, since we are not working with nucleotide bases but rather with genome order, we must also familiarize ourselves with *noise*, a concept that dates back to an interests of Einstein (1905). In the communication domain, noise (unwanted random disturbances) make it difficult to have a trivial signal. These fluctuations occur when there is a suspected origin that implicates the action of a very large number of "degrees of freedom" or variables. The coupling of noise to nonlinear systems can lead to non-trivial effects like, unstable equilibria and shift bifuracations [3]. Therefore it is widely believed that noise drastically modifies the deterministic dynamics of this system adopting its stochastic qualities. So in our case, noise behaves similarly to mutations where a singular random genomes might get flipped. The way we simulate this behavior on a computer is by random probability. Though it may be possible to destroy one of our best solutions, mutations are essential to evolution. With that being said although there is some random probability to destroy our best fit solution, there is also an equal chance that we construct an even better one.

2.4 Loss Function

Now that the algorithm's components are understood, we can also explain the calculus behind the machine learning. A *Loss Function* is what governs the learning of a machine. Its a method of evaluating how well the algorithm performs given some set of data. And in optimizations we are seeking to find the minimums of these loss functions. However, there are ways to improve the overall accuracy of the predicitions. Therefore, *Backwards Propogation*, is a mathematical tool for improving the accuracy of predictions in machine learning. As we saw in **Figure 3**, the algorithm slowly begins to pick up on accuracy at a rapid pace, even though it is a stochastic. Primarily in optimizations, finding the minimum or the local minima of a function can tell you the minimal cost of a particular function. In fact mathematically this is exactly what we are looking for. But the issues that arise is how we could have this algorithm learn to make decisions based on the data they are presented.

In vector calculus there is a concept called the *gradient* (∇), which will essentially provide the slope of a tangent plane to indicate how deep it is. This information is particularly useful to us because if we take the negative of

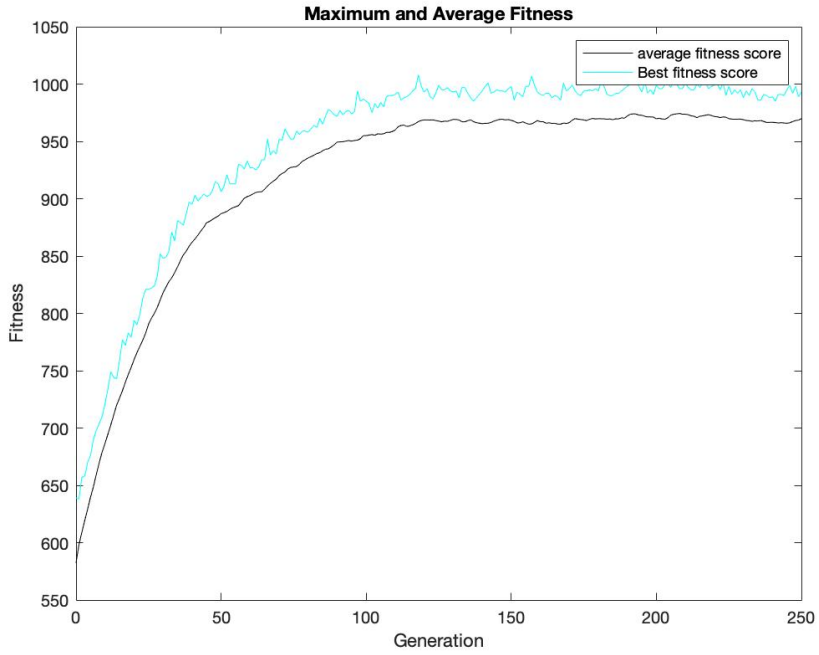

Figure 3: Graph that shows the improved fitness over a series of 250 iterations

the gradient of some cost function C, $-\nabla(C)$ [4], we will begin to know which direction to move along the plane. So as long as the negative of the gradient is being taken, we will continue to move towards the local minima of a xyz plane. This process is what we call *The Gradient Descent*. However if we refer back to **Figure 3**, we will notice that the curve will begin to plateau and sort of oscillate around this most fit region. The reason this occurs is due to the characteristics of this stochastic algorithm. Even when the algorithm obtains their "most optimal" solution, it does not know that it is the best until all the iterations have concluded. And since the single-point crossovers occur every generation, these "most optimal" solutions get altered. Therefore a better representation of our algorithm will appear as a *stochastic gradient descent*. These two concepts are pictured in **Figure 4**.

3 Phylogenetics

Phylogenetics is the study of evolutionary relationships among biological species. The similarities in biological function and molecular mechanisms in these species or *taxa* suggests that species come from a shared ancestor. This method has been used by bioinformaticians, to identify disease causing microbes, and determine its origin, how it might have spread, and routes of transmissions. As a result modeling the Delatcoronavirus variant should be no different and we wish to model an evolutionary tree with respects to time. In this temporal framework, time flows from the roots to the tips. And to completely understand the architecture of these trees we also need to introduce a little bit of graph theory.

The basic components of a phylogenetic tree as represented by **Figure 5**, contains branches, also called edges that are connected to and terminate nodes or vertices. Branches can be classified in one of two ways: internal or external (terminal). These terminal nodes that are found at the tips of trees represent *operational taxonomic units* also reffered to as OTU's. The essence of an OTU is to correspond their molecular sequences or species (taxa) from which the tree is deduced. Conversely, internal nodes represent the last common ancestor to the nodes that arise from the that point. The type of tree, we intend to work with is composed of a multi-gene family called a gene tree which looks at one species and all its different variations.

On top of the basic components of a tree we also introduce a *clade*. Clades which will appear frequently are structures of organisms that includes a single ancestor and all of its descendents [7]. Also called monophyletic groups, these represent unbroken lines of descent. An easy way to visualize this concept is to clip any single

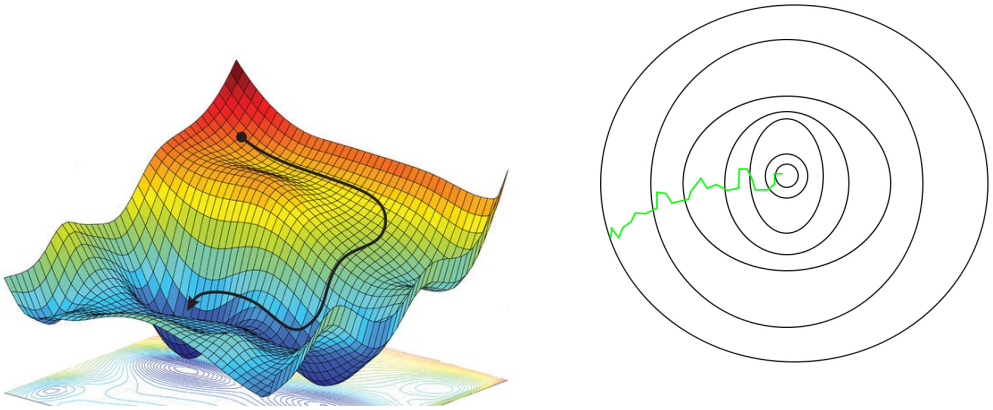

Figure 4: Gradient Descent in a xyz plane, Stochastic Gradient Descent in a depth map

branch off a tree. All the descendants on that branch is considered to be apart of that clade. If for whatever reason, there needs to be a cut made from a separate group from the rest of the tree, that group does not form a clade. Such non-clade groups are called polyphyletic. A visualization of this concept can be seen in **Figure 6**.

3.1 NP-hard Problems

Finding the most fit tree (solution) under any criterion, is considered to be a NP-hard problem within phylogenetic trees. In computer science there remains an unsolved hypothesis known as the *P versus NP problem* [5]. Computer Scientists typically like to ask "How long will it take to execute a given algorithm?" However, instead of expressing it in minutes, seconds, they give an answer relative to the number of elements the algorithm has to manipulate. As we expressed in **Section 2**, the amount of different combinations it has to go through is an overwhelming 5.386012149439842115461E214. This would obviously take a lot of time to solve, and therefore has properties of an NP-hard problem. Another way to look at this problem is to determine whether every problem whose solutions can be quickly verified, can also be quickly solved for. P stands for polynomial time, which roughly means a set of relatively easy problems to be solved for. NP stands for non-deterministic polynomial time which is a set of difficult problems to be solved for. So, if P = NP , this would imply that the difficult set of problems have relatively easily computed solutions.

While the hypothesis has yet to be proven, computation has been a powerful tool capable of solving NP-hard problems. With machine learning having a much larger presence in the way we analyze data, we could use a metaheuristics like the GA to approach these problems. So being able to solve problems that we thought could not be solved in a lifetime has opened up a new area of research and researchers are just scratching the surface in solving NP-hard problems. This research aims to solve the phylogenetic tree construction, a NP-hard problem in bioinformatics, using the GA as our focal study. The two approaches we use are the *Distance Matrix*, to provide instructions on how to build the tree and the *Neighbor-Joining Algorithm* which builds the tree itself.

3.2 Dynamical Systems

In mathematics, a *dynamical system* is a system in which a function describes the time dependence of a point in a geometrical state space. The evolution rule of the dynamical system is a function that describes what future states follow from the current state. In our case we are working with a completely stochastic system, in which random events affect the evolution of the geometric manifold. This field of study has wide applications in many biological sciences, and to an extent the phylogenetic tree is a representation of a dynamical system. Because our focus is the phylogenetic time tree the branch lengths actually have a correspondence to time. We will explore the two methods (The Distance Matrix Approach, and Neighbor-Joining Algorithms) in the coming sections and what maths are applied within these components. But before doing so, we must introduce a big aspect of phylogenetics which is analyzing the bifurcations that occur within these trees.

Phylogenetic inferences are depicted in the form of a hierachical bifurcation tree. Bifurcations, which are a series of branching processes, have one branch line and it is split into two descendents. [6] Therefore positioning

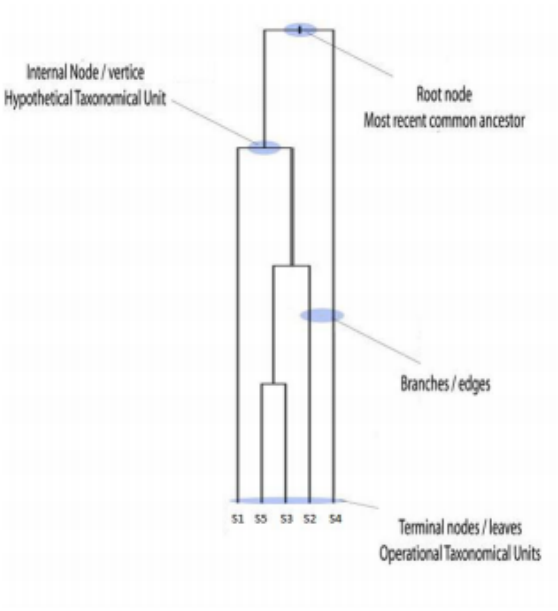

Figure 5: The phylogenetic tree structure diagram

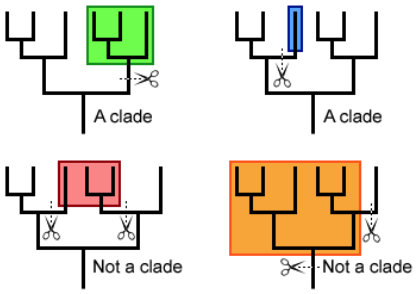

Figure 6: The distinctions of what is a clade and what is not a clade

of organisms on these trees rely solely on their genetic similarity between each other. Distance methods like the ones used in this research are based on measures of evolutionary distinctiveness between all pairs of taxa. So if we are analyzing 112 Deltacoronvirus variant genomes, there is no guarantee that all these genomes have some shared ancestor amongst each other. So while bifurcations are simple splits that occur in a tree, we may also see *outgroups*. An outgroup is a lineage that falls outside the clade being studied but is still closely related. [7] These metrics are calculated based on differences of nucleotides from DNA sequence data (provided by the NBCI). In our approach this distance method is taken care of by the GA, which learns to find the similarities and constructs an approximation with the data it is provided. We now proceed to explaining our planned methodologies to build our trees.

3.3 The Distance Matrix

Our preferred method of choice and the focus of this paper will be on the *Distance-matrix Method*. Distance-matrix methods are far more rapid compared to popular methods like the *maximum-likelihood*, and it essentially measures the genetic distances between sequences. Once these distances are found, a distance matrix will be constructed, where the distance equates to the approximate evolutionary distance. However, the challenging aspect of this computational method is constructing the matrix itself. The distance matrix can be visualized as

a $n * n$ matrix, where n resembles the number of sequences. Every row corresponds to a single sequence and every columns corresponds to the distance between the sequences. By getting some collection of n sequences, the matrix below can be constructed:

$$\begin{bmatrix} d_{11} & \cdots & d_{1n} \\ d_{21} & d_{22} & d_{2n} \\ \vdots & \ddots & \vdots \\ d_{n1} & \cdots & d_{nn} \end{bmatrix} \tag{4}$$

In the distance matrix above, d_{ij} can be broken down to where the distance is measured between the i_{th} and j_{th} sequences. Though this measure of weight is up to choice with methods like *Jukes-Canton and p-Distances*, we are going to be using a Genetic Algorithm, to find out the particular order. This method will be used to obtain the genetic distance from a its machine learning to help us construct our distance matrix which will then be fed into the neighbor-joining algorithm. Using this algorithm we will compute an unresolved star tree topology, and transforms it into a tree where all the branches lengths (time) are known. This method relies solely on how the matrix is constructed because there are a number of combinations that can be constructed within the distance matrices to feed the algorithm.

Distance-based methods like the neighbor-joining algorithm, are polynomial time and are quite rapid in practice. Within the distance-based methods, there are two different algorithms: *cluster-based* and *optimality-based*. The cluster-based algorithm builds its distance matrix by starting at similar sequence pairs, which are then used to construct the tree. This algorithm is found within the *UPGMA* or *unweighted pair group method using arithmetic average*, and our *neighbor joining*. Meanwhile the optimality-based algorithm compare multitudes of different tree topologies and selects the best fit by determining computing distances in the trees, and the desired evolution or *actual evolutionary distance*. This algorithm is common in *Fitch-Margoliash* and *minimum evolution* methods. But in retrospect, both these algorithms are computationally simpler than a maximum likliehood which is the conventional method used to approach these problems. However, since our interest is based on computational speed, we prefer this distance-matrix method over this conventional model.

3.4 Neighbor-Joining Algorithm

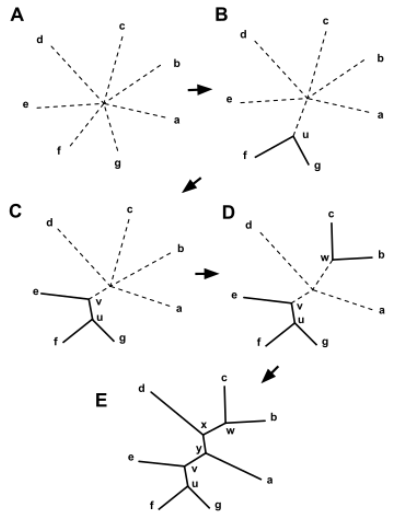

Figure 7: A visual representation of a distance matrix containing 7 species, beginning in the star-shape which then is then slowly constructed into a tree using the neighbor-joining algorithm based off a specific distance-matrix

Now that we have obtained a distance matrix, we can begin to construct the trees. As described in **Figure 7**, the tree begins in a star formation where there is no tree or lengths of branches shown or combined into internal

nodes. But with this distance matrix, the distances represent the dissimilarity of the aligned sequence and can begin subsequent joining between neighbors. So from the diagram we can see that each OTU is represented as a fork. For each OTU, you want to compute the sum of distance between OTU one and another OTU, divided by (N-2), where N is the total number of OTU's. The equation can be represented to describe the following:

$$S_i = \frac{1}{N-2} \sum_{k=1}^{N} D_{ik} \tag{5}$$

Once you combine the two taxa on a node into a sub-tree, you want to calculate the new branch length which can be calculated by the following equation:

$$D_{Xi} = \frac{D_{ij} + S_i - S_j}{2}, D_{Xj} = \frac{D_{ij} + S_j - S_i}{2} \tag{6}$$

And at last calculating the new matrix distance is done by connecting i and j and replace it with a node x that connects it to the final equation:

$$D_{xk} = \frac{D_{ik} + D_{jk} - D_{ij}}{2} \tag{7}$$

This sequence of steps are repeated for (N-2)! iterations. This mathematical algorithm is what gives the layers within the trees. After combining similar trees to each other, the matrix size begins to become smaller and smaller and the tree is built into its final form. A basic demonstration is provided in **Figure 8**

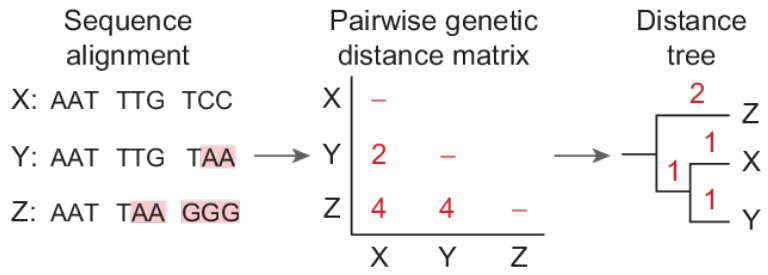

Figure 8: Basic Graphic of what is going on in the Neighbor-joining algorithm and its Distance Matrix

4 Methodology

Now that we are familiar with all the components to this tree, we can finally begin our tree construction. As mentioned frequently, the taxa we will be working with is the Delta coronavirus variant. The inspiration for modeling this taxa was the recency bias of this global pandemic. With the whole world being affected by this destructive virus, we are now beginning to see a wide array of mutations leading to different variants. Tracking 112 different genomes, we looked specifically at the spike protein to construct our tree. Our intentions to construct the tree at the protein level was to significantly reduce the runtime for our computer program. With sequence alignment techniques varying at the DNA and protein level, we decided to try a novel approach using this Genetic algorithm at the protein interface. Next we had it run through our Python program for a few minutes in which we were able to obtain the distance matrix. After doing so we fed our program to our MATLAB neighbor-joining algorithm where we are given our desired phylogenetic time tree. *For those intersted and who might have missed it, we will be citing the Github for this research at the top* .

4.1 Our Time Tree

Figure 9: Our 112 Species Time Phylogenetic Tree of the Deltacoronavirus variant.

After countless of hours of figuring out the best approaches to this problem, we have finally been able to produce a result. The measure at the bottom is equivalent to 0.09 which is around 10.8 months. For every 0.1 in length it is equivalent to 1 year. To our surprise, the Delta variant has been around for much longer than we expected. Originating from the Middle East we ware able to observe some of our early findings of this longspread disease. While we did have some surprises, we can also see on the right hand side of the time tree all the recent chaos of the Delta variant spreading across species.

4.2 Analysis and Findings

In this section we will begin discussing our analysis, and findings of the SARS-CoV-2 Deltacoronavirus and how it spread amongst the 112 species sequences offered by the NCBI. As mentioned in previous sections, we are working with this taxa at the protein interface. Therefore we display below in our table how our data was read. Obviously the spike protein sequences are much longer in length but we decided to slice the first 25 characters for the selected Delta variant genomes.

Genbank Code	Spike Protein (first 25 Amino Acids)
HKU11-934	MVKNVSKRSPIVLPQIQPPPLQLFI
HKU12-600	MAMNIAKRSPIVLPQIQPPPLQLFI
HKU13-3514	MAKNKEKRSPIVLPELVPPPLQLFI
HKU16	MGKNNPKRSPIVLPDPIPPPLQLFI
HKU17	MAKNKSKRDAIALPENVPPPLQLFI
HKU18	MAKNKEKRSPIALPEVVPPPLQLFI
HKU19	MGSKQVDHTCLTIPPNPSKTLALFI
HKU20	MANKARPKGILVIPELSNNSSLLLL
HKU21	MTKNSFDVGKVITLPKVIPPPLQLF
...	.....
HKU15	MAKNKSKRDAIALPENVPPPLQLFI

In the above table, we are working specifically with the spike proteins which show significant signs of similarities. Within these similarities are slight differences which will all be accounted for when the Genetic Algorithm program took place. But through conducting some initial analysis the similar protein sequences will be joining together first when the neighbor-joining algorithm takes place, and we can begin to see some correlations between sequences. By no means is this a surprising observation considering we are working with the same species. But it is also interesting to see how these genomes have significantly mutated over time.

Now we spend more time on the tree itself. With some of our genomes beginning at 2014, it makes intuitive sense as to why the graph is divided up as it is. With the majority of the tree falling between the 2019 to 2021 interval, it make aligns with the media and how this Coronavirus variant has spread within our "pandemic years". However within the trees there are also a few incorrect approximations. Because this Genetic algorithm only runs so many times, and is a stochastic machine learning algorithm, this randomness does make an appearance in the trees. With some genomes clearly labeled from 2014, 2016, 2018, they have been incorrectly placed towards where the majority of the data lies. There could be many reasons for these occurences. One of the reasons I could think of first off is our small sample size of genome data. The Spike protein coming from the SARS Delta Coronavirus is roughly 1,160 amino acids long. In retrospect the DNA sequence for one of these species is roughly 26865 nucletides long. Obviously if we were to run this experiment with DNA, our machine learning algorithm could pick up more insight based off the data it is presented. But as described one of of our main goals in this project was efficiency and computing time. Another possible reason for this error could be that the sequences are just very similar to one another. After running a more traditional multiple sequence alignment, we obtain the following results:

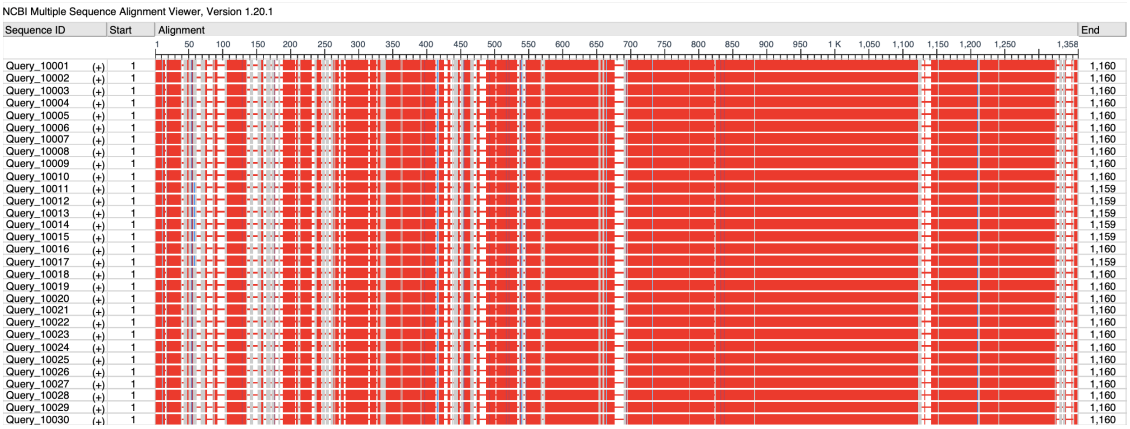

Figure 10: The first 30 genomes of a MSA.

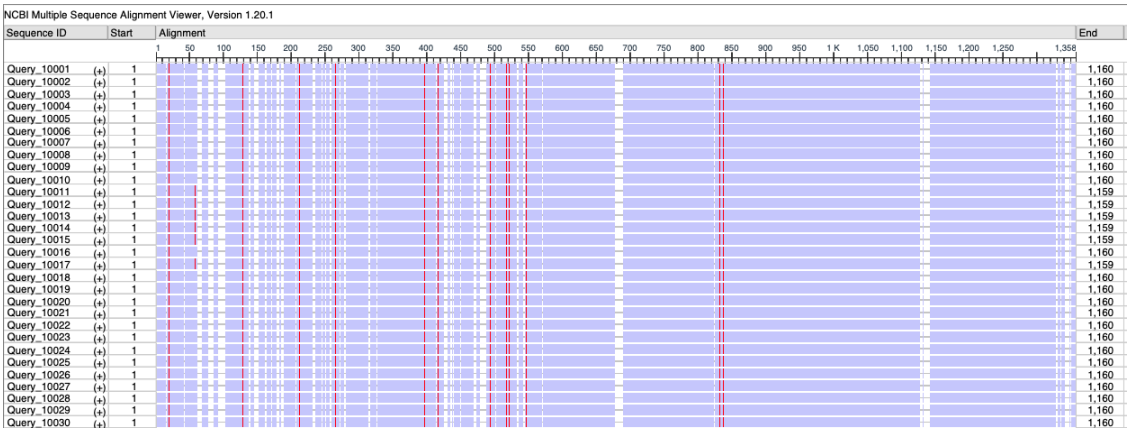

Figure 11: This visual makes it easy to see the similarities shown in **blue** and the dissimilarites shown in **red**

What I am trying to show is that throghout these genomes, they are very similar amongst each other. There are few instances where their amino acids are varying but for the most part this information is consistent. Once again this makes intutitive sense considering that these come from the same taxa.

And lastly this algorithm is indeed stochastic. Even when we may have had a better tree construction in place, there is a small chance that the tree could have been eliminated by mutations. There is no way in this algorithm to preserve the best solutions because of the single point crossovers that must take place. Even with elitism incorperated, we only select the top 3 solutions to make it to the next generation. And because single point crossovers tend to produce better solutions, this behavior fluctuates. It is for these reasons that our tree is indeed a little faulty. However, overall the representation is true to what we were expecting. There are plenty of clusters appearing as we move from left to right in the dimension of time (t). So this time tree I believe constructs a very abstract yet viable representation of a dynamical system. Given some moving variable t, our tree is bifurcating (spreading). Though this dynamical system resembles chaos moreso, with its unpredictable behaviors of when it will bifurcate, it still has many features which make it so. Though ambitious, we are content with the results we obtained from this research. Using the Genetic Algorithm as our main driver, we know the power that these machine learning algorithms can have in analyzing genomics data.

5 Conclusion

Overall this research was quite the learning experience. Not only were we able to construct a fairly accurate phylogenetic time tree, but we were also able to solve a NP-hard bioinformatics problem. With data sciences

becoming fairly impactful in biological research, we believe this is only the beginning for future advancements in bioinformatics. Slowly understanding genomics data, these phylogenetic trees can play much bigger roles in how we analyze spread of disease, and evolutionary relationships amongst different species. But with the SARS-CoV-2 pandemic with no end in sight, this field is crucial in understanding how much more dangerous this virus can potentially be.

6 Acknowledgement

I want to thank the UCSC Applied Mathematics Department for supporting this project. Having spoken about this with Professor Marcella Gomez, and Vanessa Jonnson, I was finally able to find the right time and place to conduct this research. Before this iteration of this project went into full force, my intial proposal was deemed to simple, or too easy for any time of traction to come from this research. So I thank these professors for their educational criticism. I also want to thank Professor Daniele Venturi for teaching Applied Dynamical Systems. Before taking this course I did not think I would find the connections between my project at these dynamical systems. It is for this reason that I was able to complete this project.

References

[1] Fliesler, N., *Sturdier spikes may explain SARS-CoV-2 variants' faster spread*, March 2021, Boston's Children Hospital, https://answers.childrenshospital.org/sars-cov-2-variants-spike, Accessed October 7.

[2] Ahn, C.W., Ramakrishna, R.S., *Elitism-based compact genetic algorithms*, IEEE Transactions on Evolutionary Computation, Volume: 7, Issue: 4, Aug. 2003, Pages 367-385.

[3] Gammaitoni L., Hänggi P., Jung P., and Marchesoni F., *Stochastic Resonance* Rev. Mod. Phys. 70, 223–288 (1998)

[4] 3Blue1Brown (n.d.) Neural Networks [What is backpropagation really doing? | Chapter 3, Deep learning]. Retrieved from https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

[5] Hardesty, L., *Explained: P vs. NP, The most notorious problem in theoretical computer science remains open, but the attempts to solve it have led to profound insights.*, October 2009, https://news.mit.edu/2009/explainer-pnp, Accessed October 11.

[6] *Bifurcating Trees*, Ecology Center, https://www.ecologycenter.us/genetic-diversity/bifurcating-trees.html, Acessed October 14.

[7] *Understanding Evolution: your one-stop source for information on evolution*, The Tree Room, https://evolution.berkeley.edu/evolibrary/article/0_0_0/evotrees_intro, Accessed October 14.