ViralSim - A Novel Method for Simulating Pandemics with Geographical Data

Benjamin Cai

June 2025

Abstract

Simulating real-world viral data has become integral in predicting and combating public health threats before they arise. One piece of metadata that has been overlooked though is in geography. Viruses spreading through populations are inherently based on location, meaning that an accurate method of simulating location can bring about new insights into how pandemics form along with better data to test reconstruction methods with.

Keywords: *transmission networks, transmission clusters, viral simulations, phylogenetics*

1 Introduction

Virus simulations can model the transmission, mutation, and control of viral pathogens through statistical and computational methods. These simulations serve as important tools in understanding epidemic dynamics, optimizing public health interventions, and forecasting future pandemics. By integrating parameters that are infeasible or unethical to test in real life, simulations create a framework to test hypotheses impossible to examine empirically. Past simulators have been widely focused on individual viral strains like SARS-CoV-2, HIV, or influenza [1] [2] [3]. Past simulations have taken into account complex aspects of real-world data like individual gene changes [4], multiple mutation models [5], and recombination [6].

Given both the limited timeframe of this project, the vastly underdeveloped skillset of the author, and computational limitations, the focus of ViralSim was placed upon simulating geographical transmission clusters accurately. Viruses spread from person-to-person, but at a higher level, they spread from community-to-community. These communities can range from the tiny – families and workplaces – to the large – states and countries. Statistical models like the coalescent have historically been the leading simulation technique, however, due to the geographical nature of the focus, an agent-based model was chosen. It's difficult to tell which method would've been more computationally efficient with larger datasets and with a forward-time simulation, but certain benefits in visualization and complex societal interactions are more readily available through an agent-based model. The major goals of ViralSim are to find new information on how pandemics start, test different pandemic mitigation techniques, and compare analysis techniques.

2 Approach

In ViralSim, there are three main entities: locations, hosts, and viruses. Locations encompass the geographical location, and can encode certain grouping behavior along with certain limitations on transmission. For example, location 0 may have quarantine flipped to *true*, meaning that transmissions happen $x\%$ less of the time. Location 0 might also have an even that happens every 100 time steps, meaning that some % of the hosts will group up every 100 time steps. In practice, these limitations can be specified in the input files. Hosts represent the individual members of the population. Hosts can be in four categories with concern to each individual viral family: susceptible, infected, asymptomatic, or removed. Hosts that are removed are either taken entirely out of the population by way of death or they gain an immunity to the virus in question.

ViralSim is also unique in how transmission is defined. Most commonly, transmission is defined by a simple statistical test, but due to the agent-based nature of this project (along with the author's lacking skillset), each individual virus mutation has a unique set of traits that determine transmission and viral behavior. The simplest example is in *infectionRate* which determines how readily a virus can infect a host. Other examples are *asymptomaticInfectionRate*, *deathRate*, *recoveryRate*, and *mutationRate*. Each virus also holds a unique sequence of nucleotide base pairs along with each viral family holding a phylogeny that represents the evolutionary history of said family.

The interactions between these three entities are where the important ideas lie. Hosts most simply move between locations at a set m rate. Each virus also has a chance to infect a host when an infected or asymptomatic host comes into contact with a susceptible host. Each time step a virus is within a host, there's a chance for a mutation to occur. Lastly, immunity can be bypassed after X amount of mutations which also in turn creates a new viral family. Truly lastly, hosts move around each time step based on set schedules or randomly (depending on computational resources).

The mutation types implemented in this project were substitutions, indels, and recombination which happen at rates that can both be specified or placed at rates corresponding to each other (ie. *substitutionRate* >> *indelRate* >> *recombinationRate*). The actionable effect of a mutation on a virus is a random change to all of its corresponding traits. Traits are connected to each other in an equation that can be specified by the user or a default equation where $i_r = \mathit{infectionRate}$, $ai_r = \mathit{asymptomaticInfectionRate}$, $d_r = \mathit{deathRate}$, $r_r = \mathit{recoveryRate}$, and $m_r = \mathit{mutationRate}$.

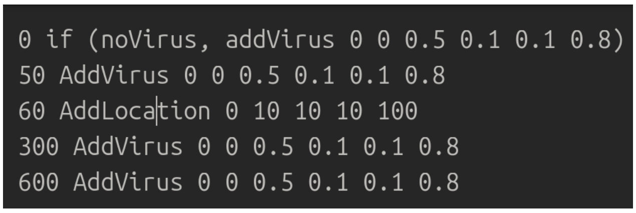
$$i_r + ai_r - (d_r + m_r r_r) = 0 \tag{1}$$

3 Inputs and Outputs for ViralSim

The main input for ViralSim are commands. The format for a command in the *commands.txt* file is of *TimeStep CommandName [Arguments]* where *TimeStep* is when the command should occur. ViralSim supports a couple of commands laid out here:

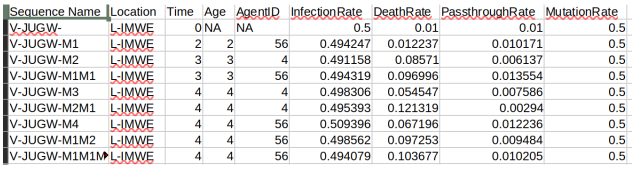
Command Name	Arguments	Effect
AddVirus	Location, Host, InfectionRate, DeathRate, RecoveryRate, MutationRate	Adds Virus to Host[*Host*] at Location[*Location*] with the given traits
AddLocation	Length, Width, Height, Number of Hosts, Max Hosts	Adds a Location with the given specifications
Masking	Location, Decrease	Flat decrease of *Decrease* to *infectionRate* at Location[*Location*]
Quarantine	Location, time	Prevents any hosts from entering or leaving Location[*Location*] for *time* time steps
if	conditional, command, Location	Given the *conditional* (noVirus, noHost, lowVirus, lowHost) at Location[*Location*], execute *command*

ViralSim outputs three main files for each viral family: sequences, sequence metadata, and the true phylogeny. Sequences is a simple multifasta file that includes every individual mutation's sequence for a viral family. Sequence Metadata holds metadata for each simulated sequence and when it was sampled (Sampling occurs randomly). The true phylogeny is the full evolutionary history of an entire viral family.



```
0 if (noVirus, addVirus 0 0 0.5 0.1 0.1 0.8)
50 AddVirus 0 0 0.5 0.1 0.1 0.8
60 AddLocation 0 10 10 10 100
300 AddVirus 0 0 0.5 0.1 0.1 0.8
600 AddVirus 0 0 0.5 0.1 0.1 0.8
```

Figure 1: *Example input commands.txt*



Sequence Name	Location	Time	Age	AgentID	InfectionRate	DeathRate	PassthroughRate	MutationRate
V-JUGW-	L-IMWE	0	NA	NA	0.5	0.01	0.01	0.5
V-JUGW-M1	L-IMWE	2	2	56	0.494247	0.012237	0.010171	0.5
V-JUGW-M2	L-IMWE	3	3	4	0.491158	0.08571	0.006137	0.5
V-JUGW-M1M1	L-IMWE	3	3	56	0.494319	0.096996	0.013554	0.5
V-JUGW-M3	L-IMWE	4	4	4	0.498306	0.054547	0.007586	0.5
V-JUGW-M2M1	L-IMWE	4	4	4	0.495393	0.121319	0.00294	0.5
V-JUGW-M4	L-IMWE	4	4	56	0.509396	0.067196	0.012236	0.5
V-JUGW-M1M2	L-IMWE	4	4	56	0.498562	0.097253	0.009484	0.5
V-JUGW-M1M1M	L-IMWE	4	4	56	0.494079	0.103677	0.010205	0.5

Figure 2: *Example output metadata*

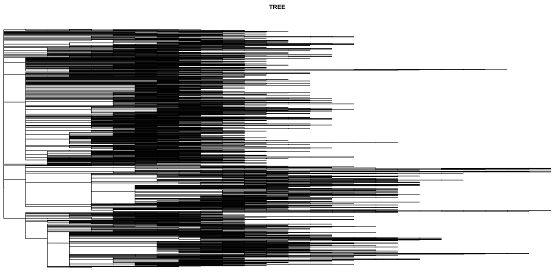


Figure 3: *Example output phylogeny*

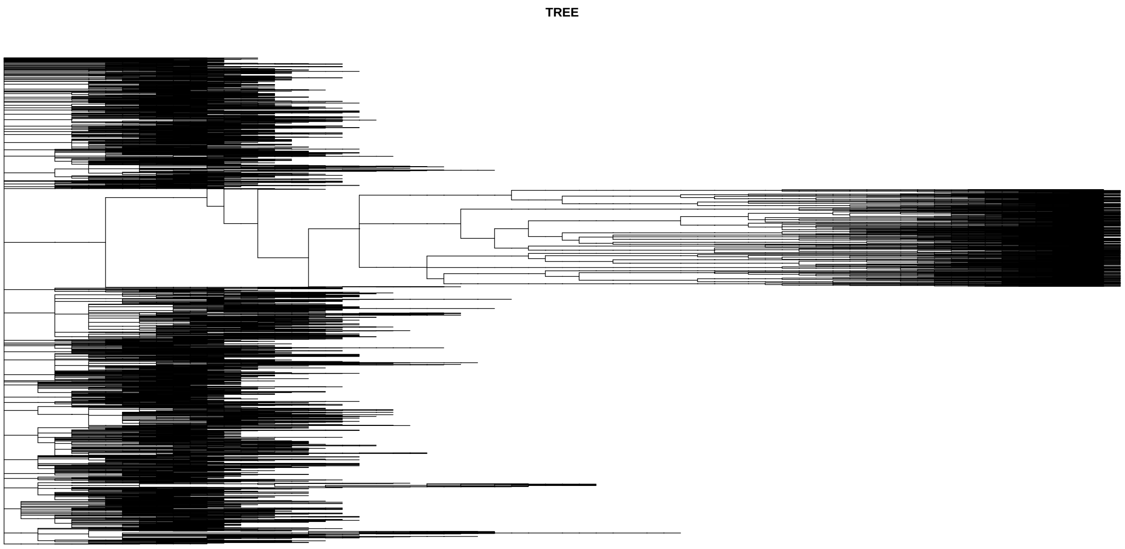


Figure 4: *Another example output phylogeny*

Something interesting is that clear clades are present in 4. Three distinct clades appear which are reminiscent of real-world viral mutations like in SARS-CoV-2 'strains'.

4 Annotating phylogenies for location

In order to discern more information from the generated data, it'd be best to understand how location relates to phylogenies. Transmission clusters are on way to understand location within viruses. Transmission clusters represent areas (locations: communities, cities, countries, etc.) that have genetically similar viral mutations. One method of visually representing this is to annotate each node (or parent) in a phylogeny based on the transmission clusters it produces. We can do this in four steps:

1. Start at a parent node with children (children representing individual sequences sampled)
2. Find the ratio of children (of parent) from location A against the number of children (of parent) not from location A
3. Any ratio above a certain threshold (say 100:1) is considered a transmission cluster
4. Repeat for all parent nodes

This idea can also be represented in an equation

$$t_c = \frac{\sum_{n=0}^{k} \begin{cases} 0 & !locationA \\ 1 & locationA \end{cases}}{\sum_{n=0}^{k} 1} \tag{2}$$

where the summation from $n = 0$ to k represents iterating over all children of the parent node. This gives us two things to watch out for t_{cr} (or the percent) and t_{cn} (or the number of sequences). Too low percent and the clade is not potent enough, too little sequences and the clade is too tiny to be significant.

5 Annotations on real data

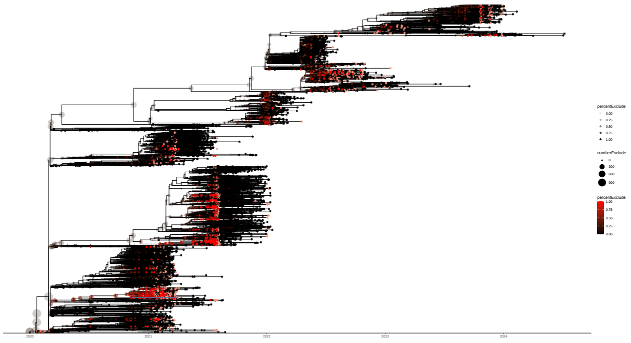


Figure 5: *US SARS-CoV-2 Phylogeny annotated for California transmission clusters*

Here's the transmission cluster identification method ran on real-world SARS-CoV-2 sequences 5.

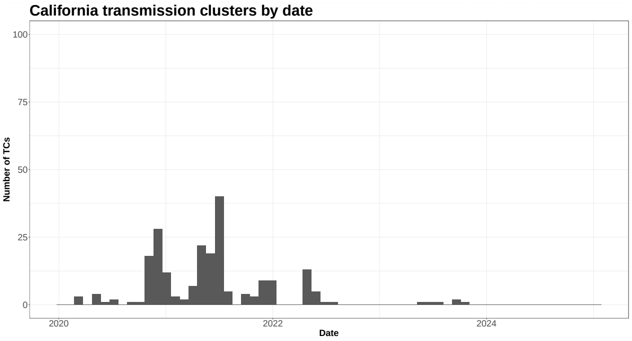


Figure 6: *SARS-CoV-2 California Transmission Clusters through time with a* $t_{cr} = 30\%$ *and* $t_{cn} = 5$

Annotating for time reveals greater insights into how the SARS-CoV-2 pandemic moved through California 6.

6 Example Annotations

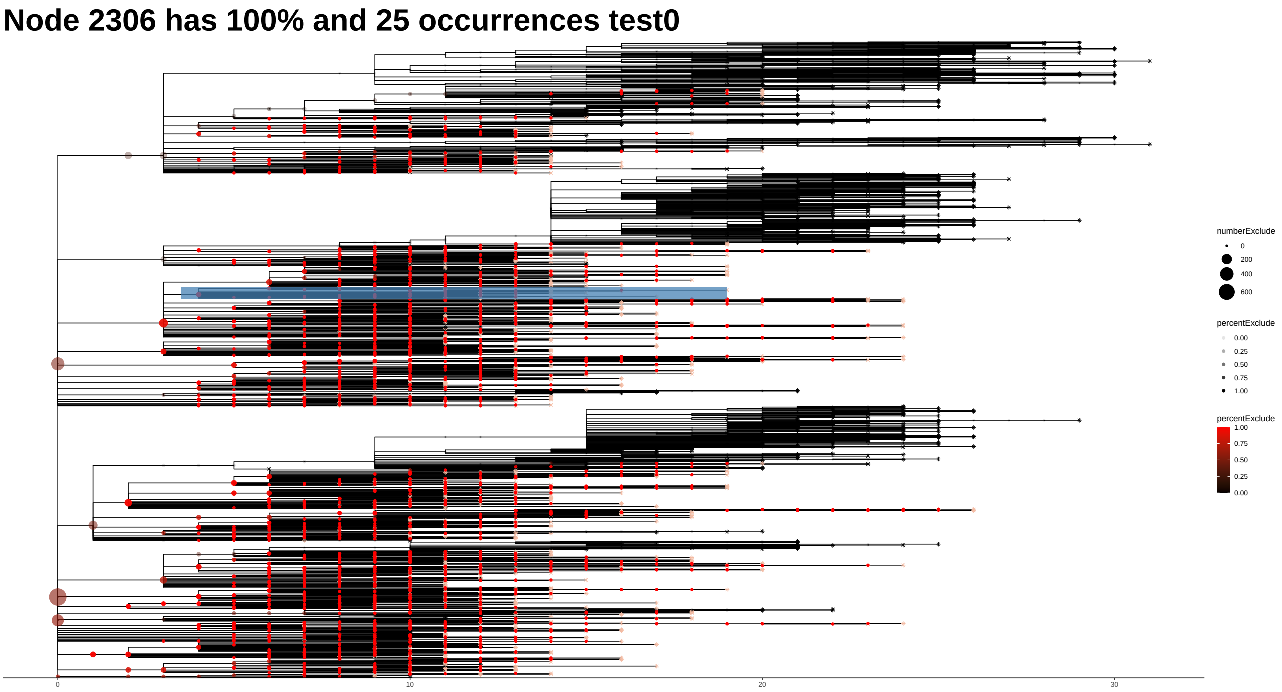


Figure 7: *One example annotation on simulated data for location test0*

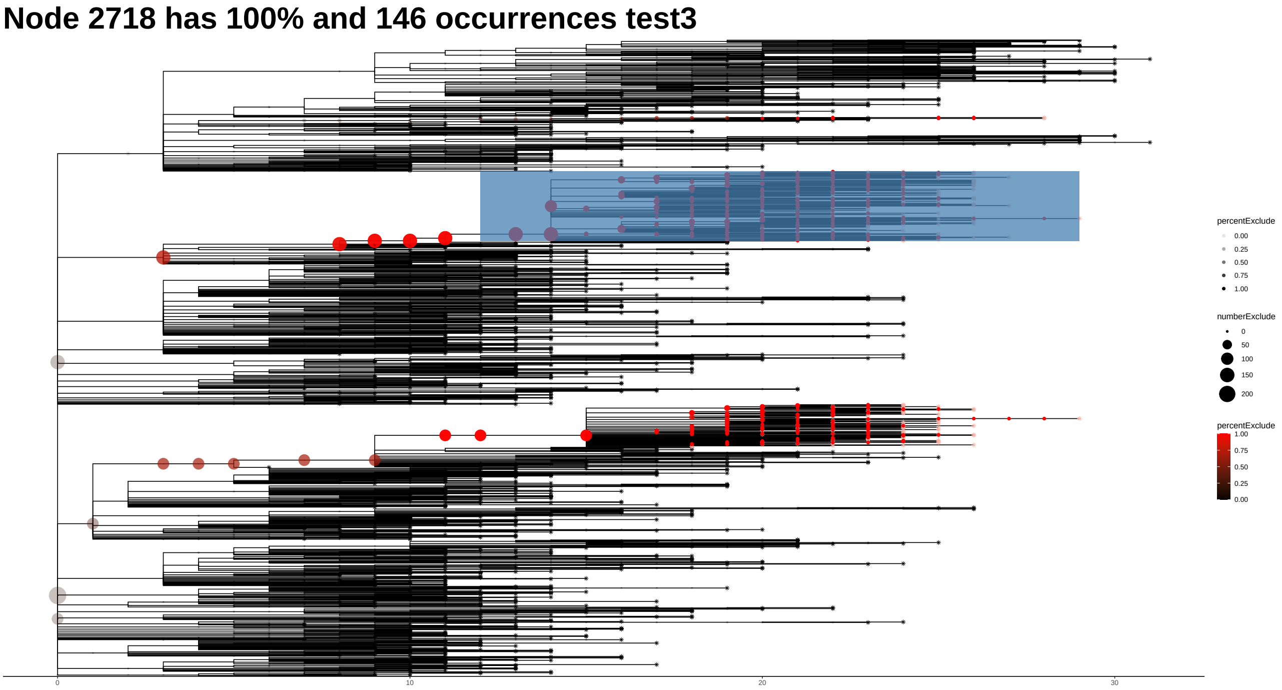


Figure 8: *One example annotation on simulated data for location test3*

Clear transmission clusters are seen in the simulated phylogenies (which is what we should be seeing).

7 Testing with real-world reconstruction techniques

Given simulated sequence data, we can reconstruct an example phylogeny to test the accuracy of those reconstruction methods. For example, faster methods like UShER [7] boast much faster speeds for greater datasets, but oftentimes have slight accuracy drawbacks.

In this case, only maximum likelihood was tested though.

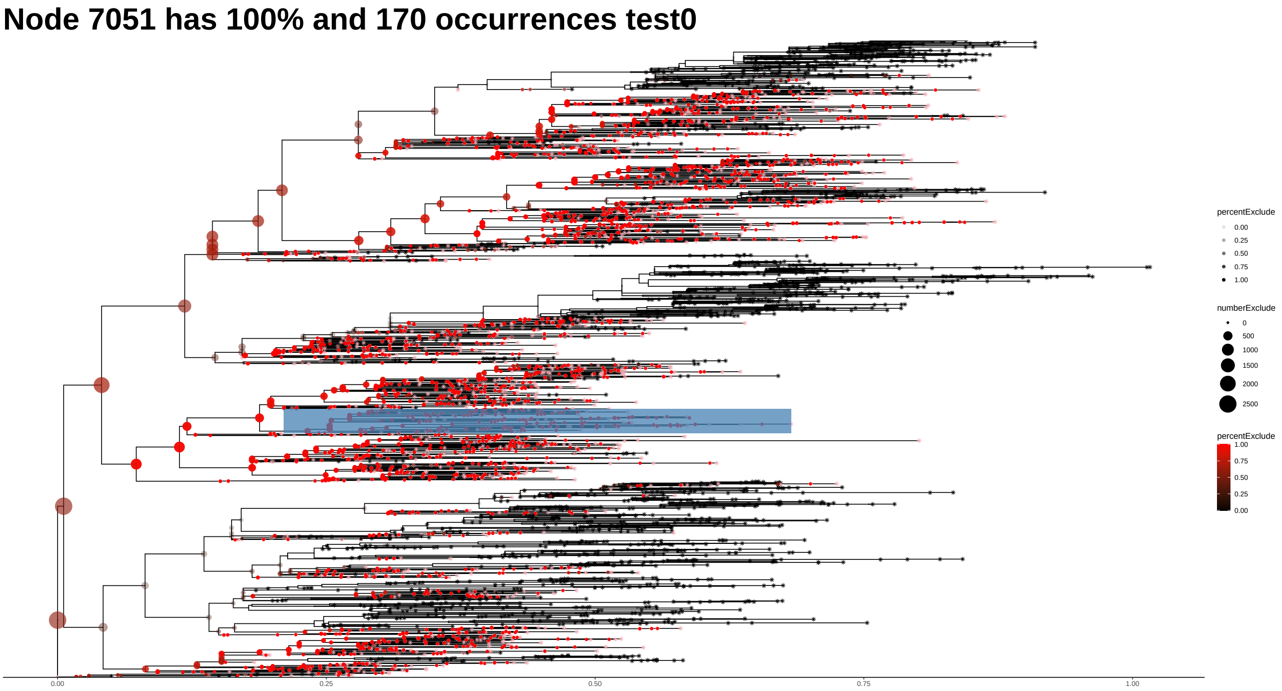


Figure 9: *One example annotation on a maximum likelihood phylogeny from simulated data for location test0*

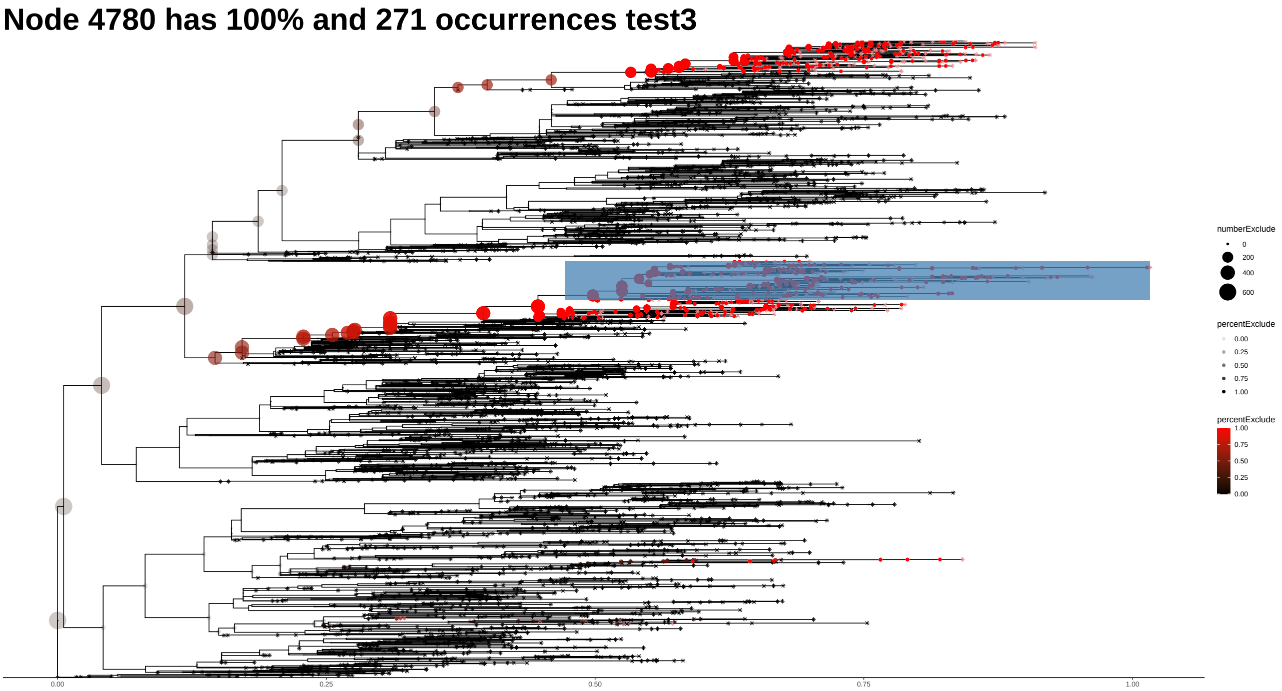


Figure 10: *One example annotation on a maximum likelihood phylogeny from simulated data for location test3*

9 should directly correspond to 7 while 10 should directly correspond to 8. If we look at the overall topology, visually, they match. More importantly though, the main transmission cluster annotations are all in the same location (7 and 8 are structurally the same if the two clades are just switched around which has no significant impact on topology).

It can also be noticed how the time steps within the simulated data phylogenies have extremely discrete time steps. This is because of the relative significance and computing power within each time step (and a possible area for improvement). While less discrete time steps are possible, they necessitate lower actions during each individual time step which may not be beneficial for the simulation trying to be run.

Something else that was tested was the impact of sampling on the overall tree topology. In real-world phylogenetics, it's impossible to have every individual mutation that occurs for a virus. This is not only because there are multiple viruses within each person when someone gets infected, but also because each individual person infected will not always report it or go to a sampling center. Therefore, it's imperative to understand how sampling (even absurd sampling rates) affect phylogenies.

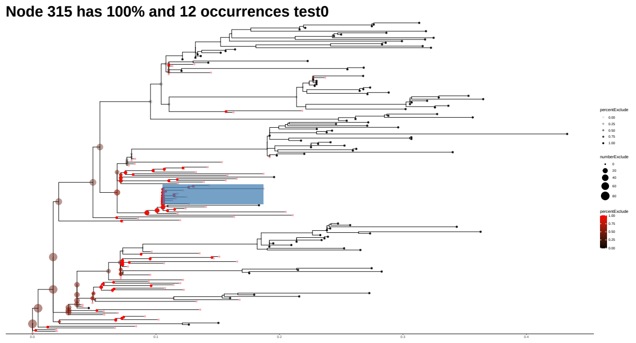


Figure 11: *One example annotation on maximum likelihood phylogeny from sampled (20%) simulated data for location test0*

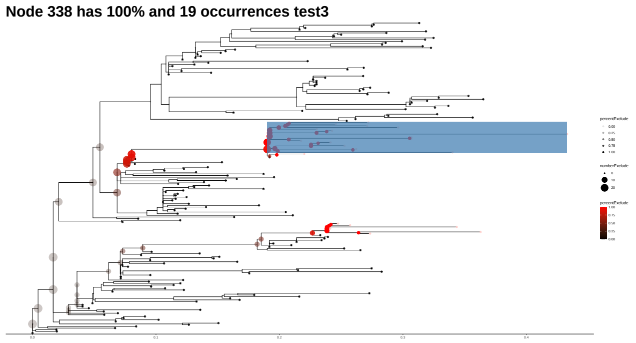


Figure 12: *One example annotation on maximum likelihood phylogeny from sampled (20%) simulated data for location test3*

In this case, sampling does maintain the topology along with overall transmission cluster potency (again for 12, 8, and **??**, imagine switching around two branches where switching branches does not impact the phylogeny significantly).

8 Fitness

Another idea that's available to ViralSim is easy access to fitness calculations. Fitness, in this case, is defined as how well a virus spreads. One plausible way of calculating fitness can be given here:

$$F = \sum_{n=0}^{k} \frac{1}{d} \tag{3}$$

where F is fitness, the summation again represents iterating over all children, and d is the distance from parent to child. The idea is that the further a child is from the parent, the less the parent's traits contributed to the child's success. The more children a parent has, the more successful or fit the parent is. Fitness in this case is directly found from the phylogeny which forgoes needing metadata or sequences.

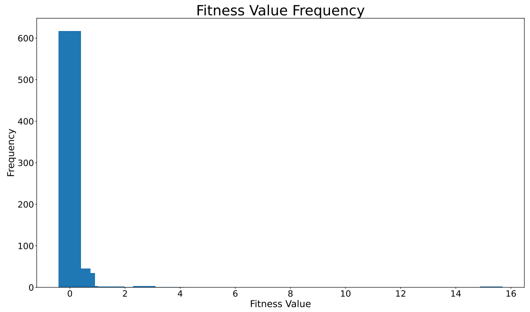


Figure 13: *Graph of fitness against frequency*

13 shows what's expected. Lots of unfit viruses.

9 Next steps

Given the time constraints, it was difficult to run more extensive benchmarks. It'd be nice to use actual quantifiable differences between phylogenies like Robinson-Foulds distances or Branch Score differences.

Also, the fitness idea is something that could be improved by trying to connect actual fitness values to the traits developed earlier and unique to ViralSim. Perhaps an explicit formula could be worked out or a model to predict fitness.

10 Acknowledgements

I'd like to thank Emma Goldberg for her help throughout the beginning of this project as well as the Institute for Computing in Research. Also, I'd like to thank Mr. Shockey's Independent Study class for their help through the various coding problems.

References

[1] “Canalization of the evolutionary trajectory of the human influenza virus - BMC Biology — bmcbiol.biomedcentral.com.” https://bmcbiol.biomedcentral.com/articles/10.1186/1741-7007-10-38. [Accessed 13-06-2025].

[2] “Mathematical Models for HIV Transmission Dynamics: Tools for Social and Behavioral Science Research — pmc.ncbi.nlm.nih.gov.” https://pmc.ncbi.nlm.nih.gov/articles/PMC3387534/. [Accessed 13-06-2025].

[3] “Modeling the SARS-CoV-2 epidemic and the efficacy of different vaccines across different network structures — journals.plos.org.” https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0325129. [Accessed 13-06-2025].

[4] “CastNet: a systems-level sequence evolution simulator - BMC Bioinformatics — bmcbioinformatics.biomedcentral.com.” https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-023-05366-1. [Accessed 13-06-2025].

[5] “INDELible: a flexible simulator of biological sequence evolution - PubMed — pubmed.ncbi.nlm.nih.gov.” https://pubmed.ncbi.nlm.nih.gov/19423664/. [Accessed 13-06-2025].

[6] “SANTA-SIM: simulating viral sequence evolution dynamics under selection and recombination — pmc.ncbi.nlm.nih.gov.” https://pmc.ncbi.nlm.nih.gov/articles/PMC6407609/. [Accessed 13-06-2025].

[7] “Ultrafast Sample placement on Existing tRees (UShER) enables real-time phylogenetics for the SARS-CoV-2 pandemic - Nature Genetics — nature.com.” https://www.nature.com/articles/s41588-021-00862-7. [Accessed 15-06-2025].