



Northeastern University

College of Engineering

VIRAL EVOLUTION

INFO 6205 Summer 2021 Project

Team Members:

Sayali Mahajan (NUID: 001576540)

Mrigesh Dasgupta (NUID: 001586781)

Akshay Bhosale (NUID: 001001091)

Instructor:

Prof. Robin Hilyard

Date: 08/13/2021

Information Systems Department

INFO 6205 50494: Program Structure and Algorithms – Section 01,
Summer Full 2021

Introduction:

The virus in its natural state is a submicroscopic non-cellular organism that is made up of protein and nucleic acid. They are the basis of many human diseases, like MUMS, HIV, POLIO, INFLUENZA, and let us not forget the one that has been in the news and gained immense popularity in the last couple of years SARS-CoV-2. Viruses infect all living creatures for the process of reproduction. The virus does not contain enough nucleotides and hence takes advantage of the infected hosts DNA for reproducing its RNA.

Aim:

The aim of this project is to study the evolution of variants of a positive-sense single stranded RNA virus like SARS-CoV-2. Since viruses have only a Single strand of genetic material, the reproduction process is entirely dependent on the infected hosts where-in the RNA of the virus gets reproduced by the human hosts DNA repetitively and is just a simple recombination task (RNA Virus Recombination), which in rare cases may lead to shuffling errors causing Mutations in the existing virus structure. Keeping the structure of SARS-CoV-2 in mind, 30000 bases distributed in 10 individual genes, we simulate the virus reproduction in a series of hosts on an island. The hosts are sub-categorized depending on the infection status and 4 different genotypes, which will further affect the fitness value of the infecting virus.

Approach:

- Our simulation encompasses a period of 2 years.
- The SARS-CoV-2 virus has 10 genes consisting of total 30000 bases, therefore we have taken 10 individual genes together as a String, each individual character carrying the weights and probability of the individual genes.
- The DNA & RNA structure of organisms consists of A, C, G, T codons and U codon occurs only in RNA structure. For experimentation purposes, each of the regularly occurring codons has been given the same weights i.e. 10, while the codon U weighs 100 to indicate occurrence of mutation.
- As mentioned earlier, the 10 genes are represented as a string of 10 characters (ABCDEFGHIJ). Where the probability of mutation is proportional to the length

Viral Evolution Simulation

of the gene and therefore whenever a mutation occurs in a gene, the character is replaced by a number to indicate the count of mutation.

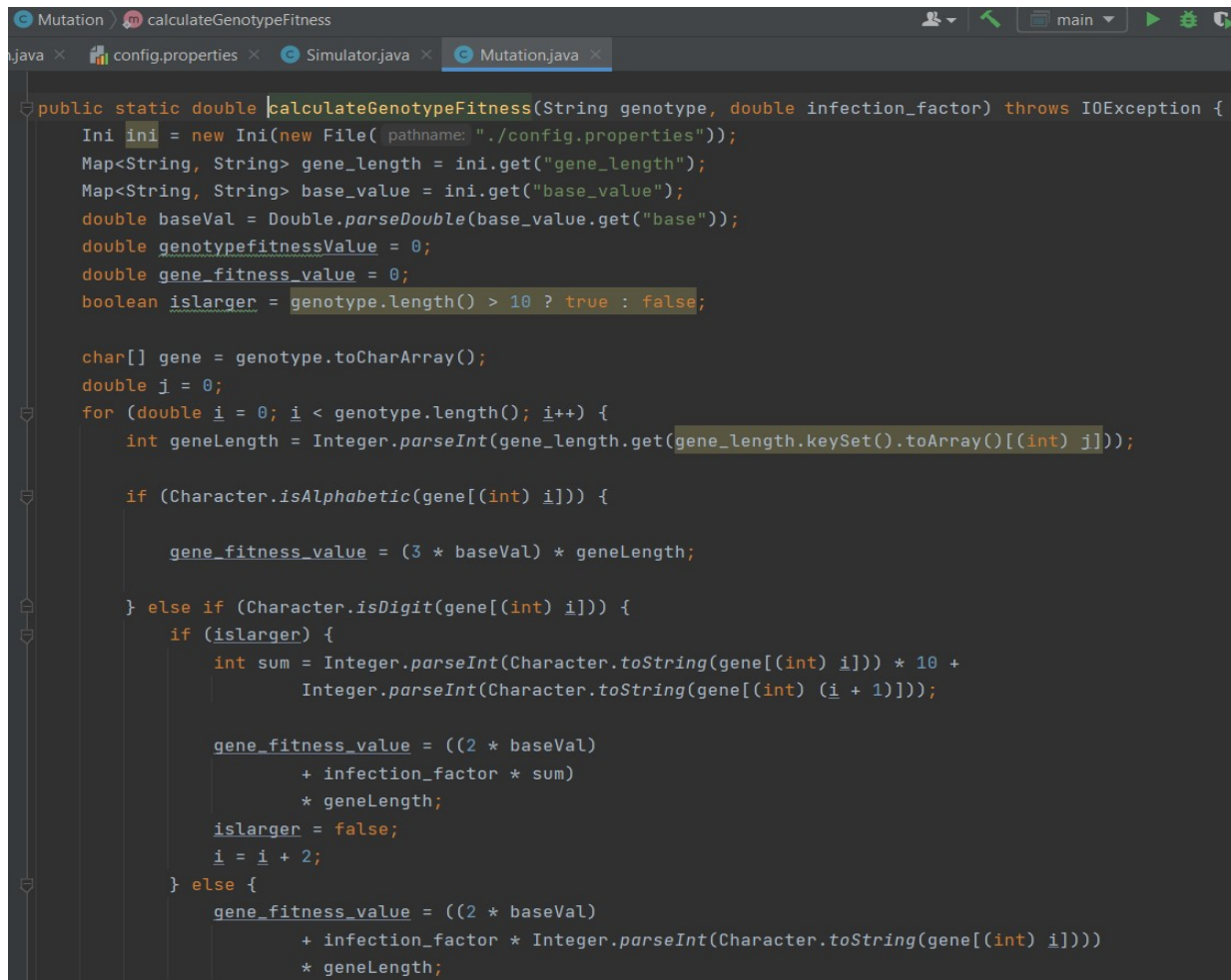
- There are 1000 host on the island for the simulation, divided between 4 different genotypes. The simulation picks 2% of the population randomly and infects them with the virus.
- The recovery period is 21 days, so all the hosts will be infected for equal period irrespective of their genotypes.
- Using the assigned weights of each gene represented in the string, we sum up the weights of the entire string calculating Fitness for each progeny of the virus.
- Vaccination of the population on the island starts after first year i.e. 365 days.

Program:

Data Structure & Classes:

Following data structures and classes are used:

- Island: This class defines the islands 2D structure for the simulation, namely the center and the boundary are defined for establishing RandomWalk algorithm.
- Mutation: This class Calculates Genotype Fitness for each mutation of virus. Apart from that it also consists of the method for calculating MutationFactor which indicates if we have a dominant variant of the Virus.



```
public static double calculateGenotypeFitness(String genotype, double infection_factor) throws IOException {
    Ini ini = new Ini(new File( pathname: "./config.properties"));
    Map<String, String> gene_length = ini.get("gene_length");
    Map<String, String> base_value = ini.get("base_value");
    double baseVal = Double.parseDouble(base_value.get("base"));
    double genotypefitnessValue = 0;
    double gene_fitness_value = 0;
    boolean islarger = genotype.length() > 10 ? true : false;

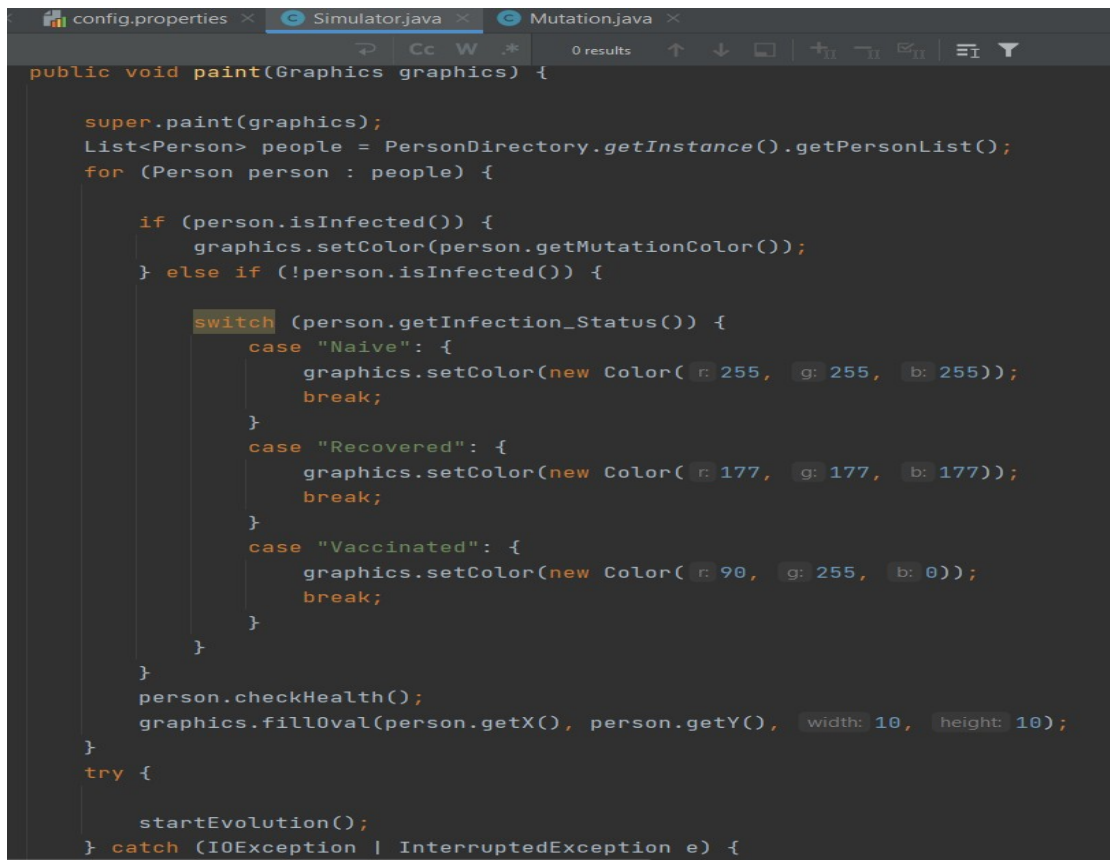
    char[] gene = genotype.toCharArray();
    double j = 0;
    for (double i = 0; i < genotype.length(); i++) {
        int geneLength = Integer.parseInt(gene_length.get(gene_length.keySet().toArray()[j]));

        if (Character.isAlphabetic(gene[(int) i])) {
            gene_fitness_value = (3 * baseVal) * geneLength;
        } else if (Character.isDigit(gene[(int) i])) {
            if (islarger) {
                int sum = Integer.parseInt(Character.toString(gene[(int) i])) * 10 +
                    Integer.parseInt(Character.toString(gene[(int) (i + 1)]));

                gene_fitness_value = ((2 * baseVal)
                    + infection_factor * sum)
                    * geneLength;
                islarger = false;
                i = i + 2;
            } else {
                gene_fitness_value = ((2 * baseVal)
                    + infection_factor * Integer.parseInt(Character.toString(gene[(int) i])))
                    * geneLength;
            }
        }
    }
}
```

Viral Evolution Simulation

- MathUtil: This class is used for random number generation for random walk, using Gaussian method of Random class under Util package.
- Person: This class calls the Random Walk algorithm moving the hosts around the island generating random moves for all hosts. This class also maintains the attributes for the hosts.
- Person Directory: This class maintains a list of the Persons class object for storing attributes of individual hosts. As well as counters of infected, vaccinated, dead and host categories.
- Position: This class contains methods that return the host positions on the X-Y axis of the island.
- RandomWalk: This class defines the attributes for our RandomWalk Algorithm, like rePositioned host locations and flag for repositioning.
- Simulator: This class maintains the chart functions to generate graphs for the JPanel, changes colors of the host on the JPanel, evolves the virus, initiates vaccination and virus mutation.



```
public void paint(Graphics graphics) {  
    super.paint(graphics);  
    List<Person> people = PersonDirectory.getInstance().getPersonList();  
    for (Person person : people) {  
        if (person.isInfected()) {  
            graphics.setColor(person.getMutationColor());  
        } else if (!person.isInfected()) {  
            switch (person.getInfection_Status()) {  
                case "Naive": {  
                    graphics.setColor(new Color(255, 255, 255));  
                    break;  
                }  
                case "Recovered": {  
                    graphics.setColor(new Color(177, 177, 177));  
                    break;  
                }  
                case "Vaccinated": {  
                    graphics.setColor(new Color(90, 255, 0));  
                    break;  
                }  
            }  
        }  
        person.checkHealth();  
        graphics.fillOval(person.getX(), person.getY(), width: 10, height: 10);  
    }  
    try {  
        startEvolution();  
    } catch (IOException | InterruptedException e) {
```

Viral Evolution Simulation

- **VirusStrainMap:** This class initializes the graphs for the JPanel, and makes their respective charts along with other attributes.
- **Main:** This is the backbone of the entire simulation, this calls the Thread functions start method as well as initiate the hash maps and charts for the JFrame to aid in creating the Graphs that provide the live stats of our simulation. It also creates the object for JPanel and sets up the JPanel background.

Algorithm:

Virus Evolution Algorithm:

1. All host are in Naive category irrespective of their genotypes.
2. Get Random 2% of the available host population and infect them with the first generation of Virus Genotype (ABCDEFGHIIJ) every day of the simulation until the virus mutates.
3. Change the value of isInfected from False to True for the respective hosts.
4. The recovery period from Virus Infection is 21 days (about 3 weeks), following which change isInfected to False.
5. Change the host category to Recovered.
6. Each virus strain infects people for 25 days reproducing inside the hosts body, after which the virus mutates. All virus strains mutate after every 25 days of simulation.
7. Update recovery days for respective hosts on regular basis.
8. If recovery days count reaches 0 then change host category to Recovered.
9. Check gene length for the genotype in config.properties and mutate a single gene randomly.
10. Probability of gene mutation is proportional to gene length. Replace the gene representing character with number in the string. (For example: ABCDEFGHIIJ mutates to 1BCDEFGHIIJ)
11. Calculate the fitness value for the new genotype, for each category and type of host and store the values.
12. Considering virus reproduction in an infected host, calculate the infection factor of every genotype of the virus for a period of 21 days using fitness function.
13. Check the difference of fitness values between predecessor and current progeny of the virus genotype.

Viral Evolution Simulation

14. Check if the value returned by infected factor for the current progeny is in the 90th percentile of the above-mentioned difference. If yes, then we have a dominant variant.
15. If the infected factor returns value in the 95th percentile for the current progeny infecting a host, then we have a dead host. Change isDead value to True.
16. After 365 days of simulation, randomly pick 0.2% of the host population update their vaccination status. IsVaccinated = True.
17. Change category of respective hosts to Vaccinated.

Invariants:

- The simulation runs for 730 days. Using “java.lang” package’s class Thread we run the simulation loops for 730 iterations, each equivalent to a day.
- The algorithm mutates the virus after every 25 days, for those 25 iterations the virus strain continues to infect the hosts and reproduce itself.
- The human host populating the island in the simulation are 1000 which are further sub-categorized according to their infection status.
- We have programmed the algorithm to randomly pick 2% of the population i.e., 20 hosts daily, and infect them with the current progeny of the virus.
- Once infected, the host shall stay infected for 21 days. So, the recovery days count is set at 21 and shall go decrementing as and when the host infection status is changed.
- The island width and height are 700×800 , which we use as upper limits for the Random Walk algorithm.
- The `dominant_factor` is 0.91 establishes the percentage to check fitness value differences between new and previous progeny of virus strains. This further comes in handy when we check host infected virus’ fitness function value. If infected fitness value is equivalent or higher than the dominant factor calculated value then we raise a flag to indicate a dominant strain of mutant virus.
- The higher limit of mutation is set to 30. Therefore, throughout the entire experiment on the simulation there cannot be more than 30 mutations.
- After 365 days of the simulation, 0.2% of the population is picked at random and their status is changed to vaccinated. As stated in the problem statement we start rolling out vaccination after a year but we deliberately chose to keep the count low otherwise the entire population would get vaccinated before the end of the experiment.
- The host population is of 2 genes, each with two alleles. The host population has 4 genomes and 3 subcategories: [Naive, Recovered & Vaccinated] x [A1, A2, B1, B2]. Therefore, we have 12 different types of hosts throughout the experiment.

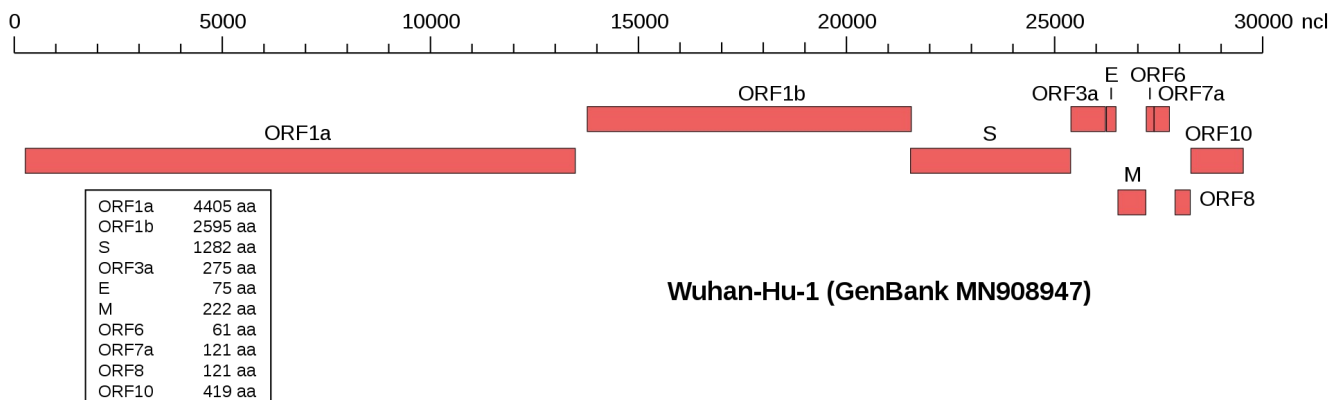
Viral Evolution Simulation

- For each category of the host, the genomes have been assigned values such that every mutation (occurrence of U=100 codon) of the virus would never be 100% effective. This is represented as their infection factor. Following is the table:

Table 1: Infection Factor for Different Types of Hosts

Category/ Genome	Naive	Recovered	Vaccinated
A1	93	86	82
A2	94	87	83
B1	95	88	84
B2	96	89	85

- As mentioned earlier in the introduction and approach, each codon base irrespective of its type has been taken to have the same weights, apart from codon U. Codon base U =100 for experimentation purposes. Now whenever a virus strain mutates, we assume the occurrence of U in that particular gene, which then further aids in calculation of the fitness value.
- We are using the SARS-CoV2 genome as our base, therefore we also have 10-character genome string, where each character represents an individual gene. Following is the SARS-CoV-2 genome:



- Now we know that the probability of a gene's mutation is proportional to its length.
- Each gene consists of triplets of codon bases. Now as we mentioned in the earlier point, if we assume each base to have a weight of 10 then every gene has a

Viral Evolution Simulation

common factor of 30 irrespective of which bases make up the gene, apart from the U codon base.

- Breaking down the above Figure 1 for better understanding of our logic:

Table 2: SARS-CoV-2 Gene Weight Distribution

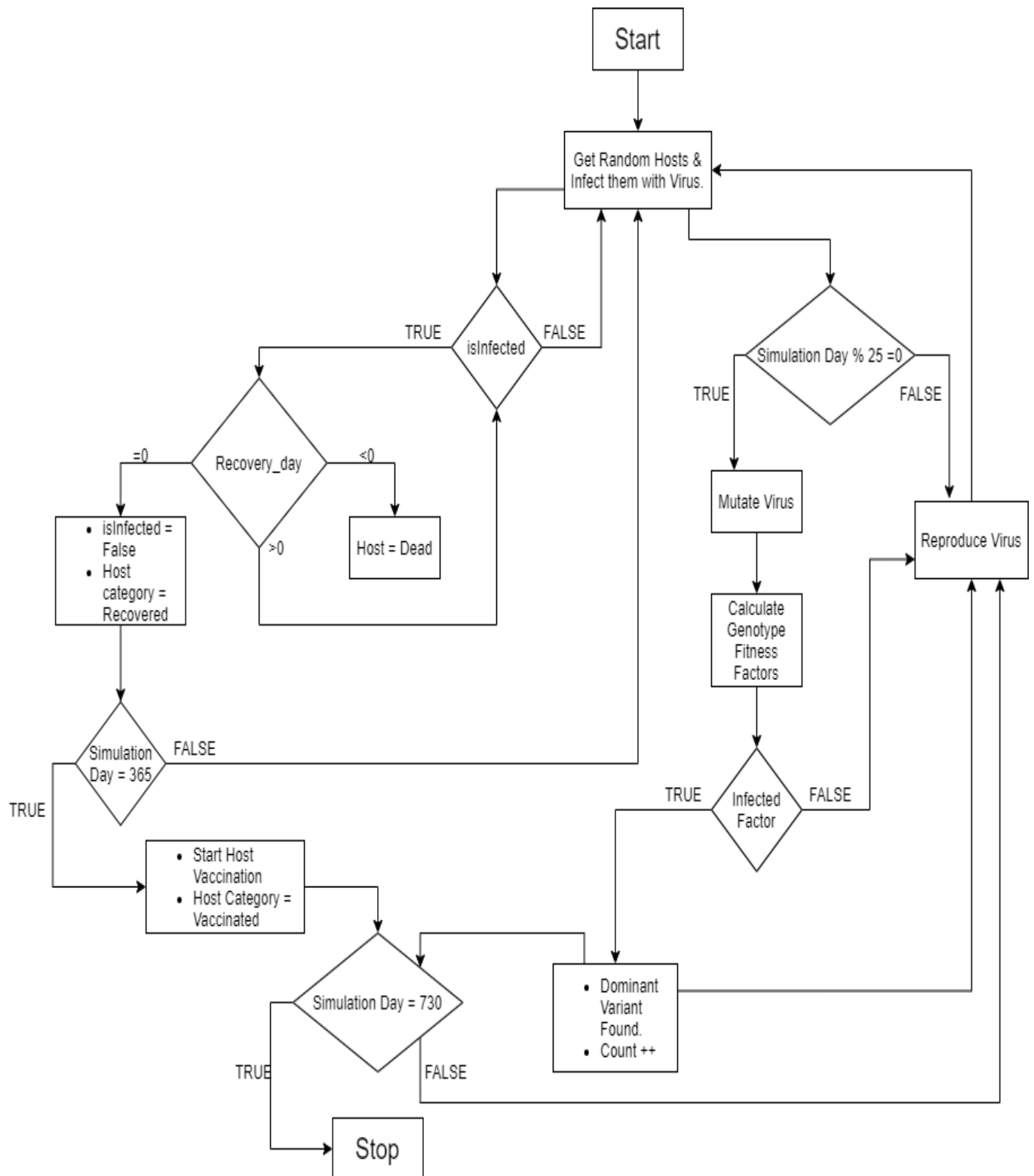
Gene	Character	Weight Factor	Length	Total
ORF1a	A	30	400	12000
ORF1b	B	30	300	9000
S	C	30	200	6000
ORF3a	D	30	30	900
E	E	30	3	90
M	F	30	20	600
ORF6	G	30	3	90
ORF7a	H	30	20	600
ORF8	I	30	10	300
ORF10	J	30	20	600

- For example, ABCDEFGHIJ has a total weight of 30180. Now let us consider that there is a mutation in gene ORF1a i.e., A. So now our string will be represented as 1BCDEFGHIJ, where 1 indicates the count of mutation in that gene. Now our total weight for the mutated gene is as following:
 - A: $[(2 \times 10) + 1 \times 100] \times 400 = 48000$
 - B: $30 \times 300 = 9000$
 - C: $30 \times 200 = 6000$
 - D: $30 \times 30 = 900$
 - E: $30 \times 3 = 90$
 - F: $30 \times 20 = 600$
 - G: $30 \times 3 = 90$
 - H: $30 \times 20 = 600$
 - I: $30 \times 10 = 300$
 - J: $30 \times 20 = 600$
 - Total fitness value = 66180
- Hence probability of mutation in gene A is highest followed by B and C, while lowest for genes E and G.

Viral Evolution Simulation

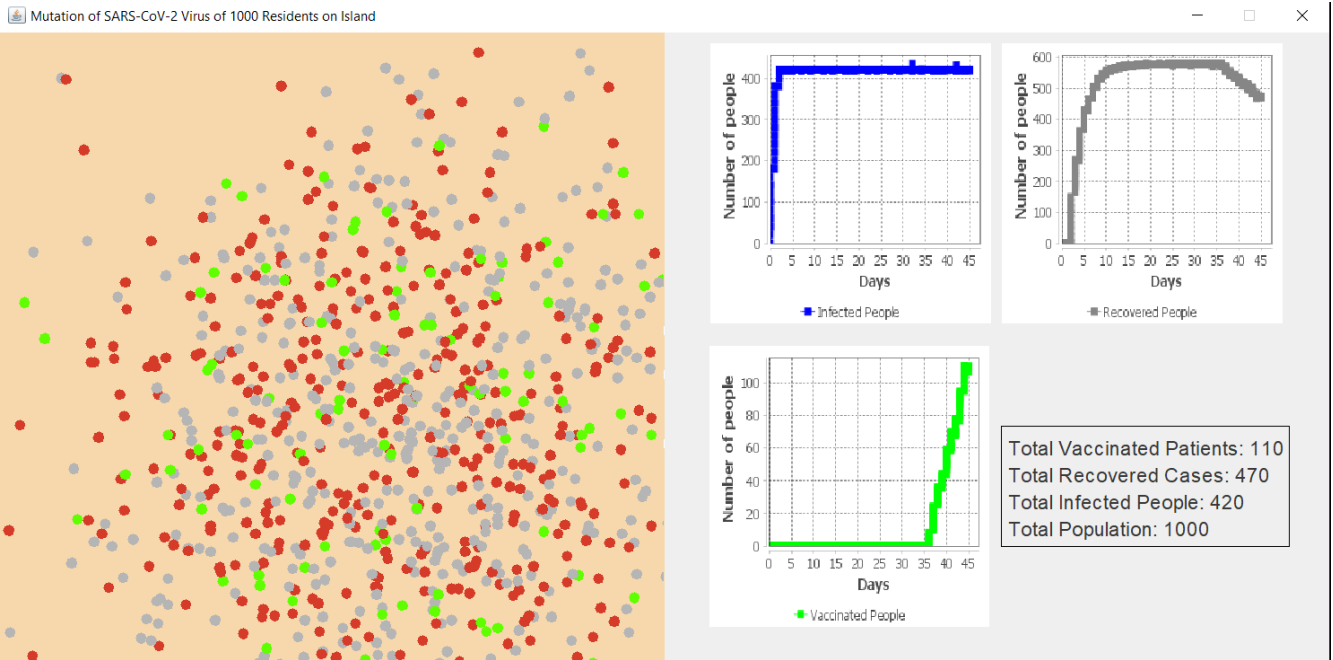
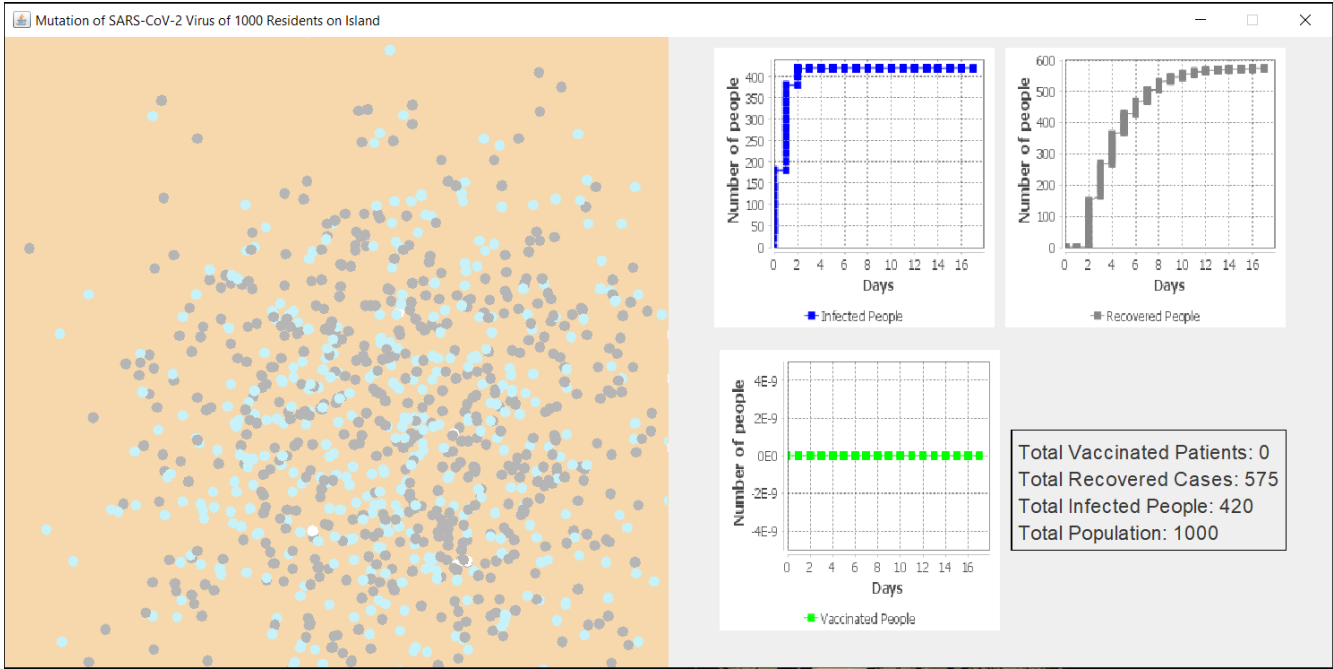
- Now let's take these two virus genotypes and check if the new mutant is a dominant variant, for explanation purposes we'll take the values of Naive A1 host:
 - Naive A1: 93
 - Predecessor Virus Fitness Value: 30180
 - Mutant Fitness Value: 66180
 - Difference between both the fitness values = 36000
 - Dominant factor = 0.91
 - Threshold for dominant variant = $(0.91 \times 36000) + 30180 = 62940$
 - Fitness value of new mutant genotype (1BCDEFGHIJ) in Naive A1 is:
 - A: $[(2 \times 10) + 1 \times 93] \times 400 = 45200$
 - B: $30 \times 300 = 9000$
 - C: $30 \times 200 = 6000$
 - D: $30 \times 30 = 900$
 - E: $30 \times 3 = 90$
 - F: $30 \times 20 = 600$
 - G: $30 \times 3 = 90$
 - H: $30 \times 20 = 600$
 - I: $30 \times 10 = 300$
 - J: $30 \times 20 = 600$
 - Total fitness value = 63380
 - Since fitness value is greater threshold calculated, therefore the current genotype is a dominant variant of the virus.

Flowchart:

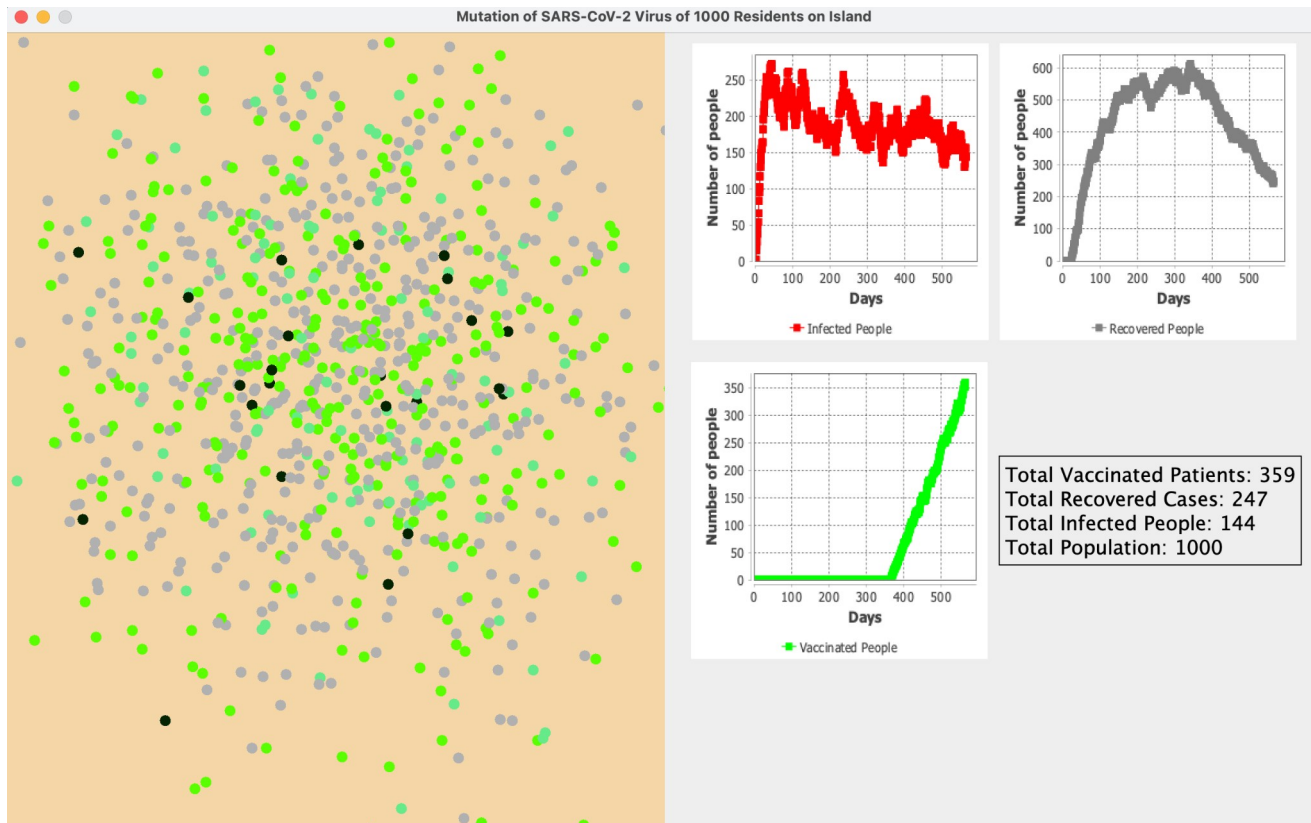


Viral Evolution Simulation

UI Flow:



Viral Evolution Simulation



- Each white dot on the simulation JPanel in the above image represents an alive host of Naive category.
- Every host getting infected with a new generation of Virus is colored with a new color representing a mutation in the existing virus genotype.
- Every host that recovers from a virus infection is turned grey in color, which indicates they are again open and susceptible for virus infection.
- The hosts that do not survive a new genotype of virus infection are indicated with black color.
- For the entire period that simulation runs on a computer, live stats of recovery, infection and vaccination are displayed using graphs with chart methods.
- Thus you can see the mathematical analysis right away in live on the JPanel as well.

Viral Evolution Simulation

Command Line Result:

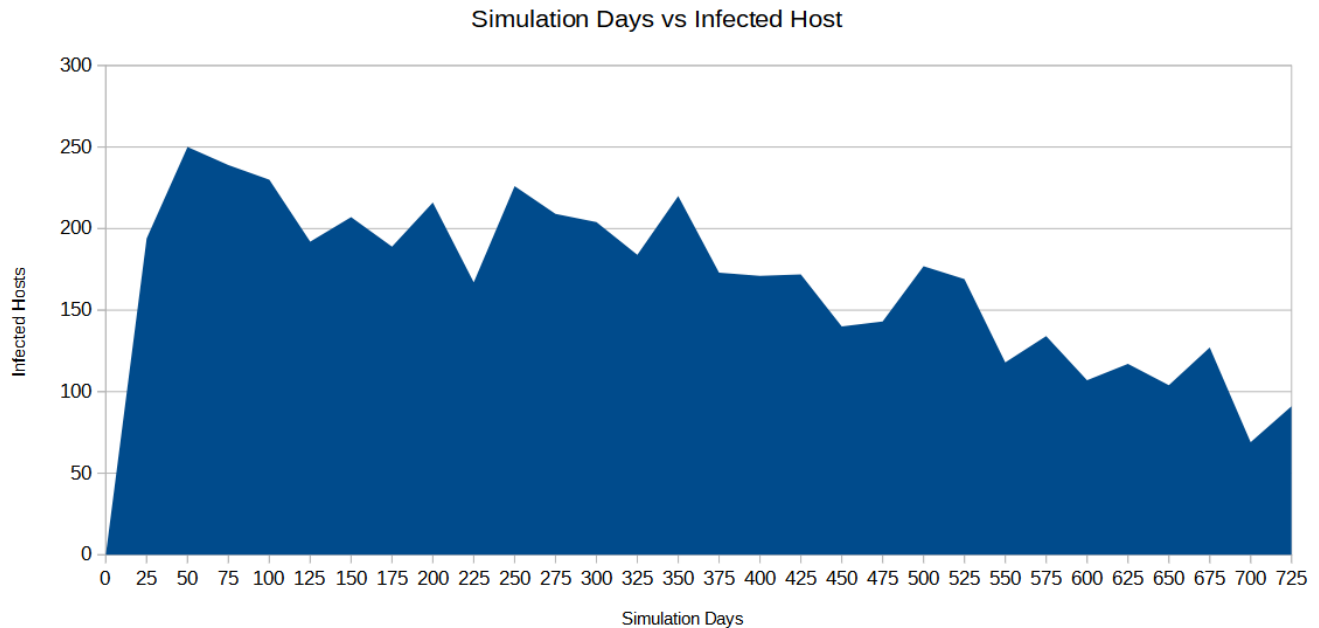
```
Mutation 15 : 422111111J is Variant : false
Day : 350 | Naive : 4 | Infected : 159 | Recovered : 588 | Vaccinated : 0 | Dead : 249
Day : 351 | Naive : 4 | Infected : 162 | Recovered : 585 | Vaccinated : 0 | Dead : 249
Day : 352 | Naive : 4 | Infected : 174 | Recovered : 573 | Vaccinated : 0 | Dead : 249
Day : 353 | Naive : 4 | Infected : 179 | Recovered : 568 | Vaccinated : 0 | Dead : 249
Day : 354 | Naive : 4 | Infected : 174 | Recovered : 573 | Vaccinated : 0 | Dead : 249
Day : 355 | Naive : 4 | Infected : 167 | Recovered : 580 | Vaccinated : 0 | Dead : 249
Day : 356 | Naive : 4 | Infected : 175 | Recovered : 572 | Vaccinated : 0 | Dead : 249
Day : 357 | Naive : 4 | Infected : 172 | Recovered : 575 | Vaccinated : 0 | Dead : 249
Day : 358 | Naive : 4 | Infected : 168 | Recovered : 579 | Vaccinated : 0 | Dead : 249
Day : 359 | Naive : 4 | Infected : 166 | Recovered : 581 | Vaccinated : 0 | Dead : 249
Day : 360 | Naive : 4 | Infected : 173 | Recovered : 574 | Vaccinated : 0 | Dead : 249
Day : 361 | Naive : 4 | Infected : 179 | Recovered : 568 | Vaccinated : 0 | Dead : 249
Day : 362 | Naive : 3 | Infected : 187 | Recovered : 561 | Vaccinated : 0 | Dead : 249
Day : 363 | Naive : 3 | Infected : 197 | Recovered : 551 | Vaccinated : 0 | Dead : 249
Day : 364 | Naive : 3 | Infected : 187 | Recovered : 561 | Vaccinated : 0 | Dead : 249
Day : 365 | Naive : 3 | Infected : 188 | Recovered : 560 | Vaccinated : 0 | Dead : 249
Day : 366 | Naive : 3 | Infected : 180 | Recovered : 566 | Vaccinated : 2 | Dead : 249
Day : 367 | Naive : 3 | Infected : 174 | Recovered : 570 | Vaccinated : 4 | Dead : 249
Day : 368 | Naive : 3 | Infected : 182 | Recovered : 560 | Vaccinated : 6 | Dead : 249
Day : 369 | Naive : 3 | Infected : 194 | Recovered : 546 | Vaccinated : 8 | Dead : 249
Day : 370 | Naive : 3 | Infected : 183 | Recovered : 555 | Vaccinated : 10 | Dead : 249
Day : 371 | Naive : 3 | Infected : 188 | Recovered : 548 | Vaccinated : 12 | Dead : 249
Day : 372 | Naive : 3 | Infected : 180 | Recovered : 554 | Vaccinated : 14 | Dead : 249
Day : 373 | Naive : 3 | Infected : 184 | Recovered : 548 | Vaccinated : 16 | Dead : 249
Day : 374 | Naive : 3 | Infected : 183 | Recovered : 547 | Vaccinated : 18 | Dead : 249
```

```
Mutation 16 : 4221111111 is Variant : true
Day : 375 | Naive : 2 | Infected : 189 | Recovered : 541 | Vaccinated : 19 | Dead : 249
Day : 376 | Naive : 2 | Infected : 189 | Recovered : 539 | Vaccinated : 21 | Dead : 249
Day : 377 | Naive : 2 | Infected : 188 | Recovered : 539 | Vaccinated : 22 | Dead : 249
Day : 378 | Naive : 2 | Infected : 199 | Recovered : 526 | Vaccinated : 24 | Dead : 249
Day : 379 | Naive : 2 | Infected : 200 | Recovered : 523 | Vaccinated : 26 | Dead : 249
Day : 380 | Naive : 2 | Infected : 209 | Recovered : 514 | Vaccinated : 26 | Dead : 249
Day : 381 | Naive : 2 | Infected : 200 | Recovered : 521 | Vaccinated : 28 | Dead : 249
Day : 382 | Naive : 2 | Infected : 189 | Recovered : 530 | Vaccinated : 30 | Dead : 249
Day : 383 | Naive : 2 | Infected : 184 | Recovered : 532 | Vaccinated : 32 | Dead : 250
Day : 384 | Naive : 2 | Infected : 171 | Recovered : 544 | Vaccinated : 33 | Dead : 250
Day : 385 | Naive : 2 | Infected : 178 | Recovered : 535 | Vaccinated : 35 | Dead : 250
Day : 386 | Naive : 2 | Infected : 169 | Recovered : 542 | Vaccinated : 37 | Dead : 250
Day : 387 | Naive : 2 | Infected : 179 | Recovered : 529 | Vaccinated : 40 | Dead : 250
Day : 388 | Naive : 2 | Infected : 171 | Recovered : 535 | Vaccinated : 42 | Dead : 250
Day : 389 | Naive : 2 | Infected : 168 | Recovered : 538 | Vaccinated : 42 | Dead : 250
Day : 390 | Naive : 2 | Infected : 154 | Recovered : 550 | Vaccinated : 44 | Dead : 250
Day : 391 | Naive : 2 | Infected : 158 | Recovered : 544 | Vaccinated : 46 | Dead : 250
Day : 392 | Naive : 2 | Infected : 163 | Recovered : 537 | Vaccinated : 48 | Dead : 250
Day : 393 | Naive : 1 | Infected : 169 | Recovered : 530 | Vaccinated : 50 | Dead : 250
Day : 394 | Naive : 1 | Infected : 153 | Recovered : 543 | Vaccinated : 53 | Dead : 250
Day : 395 | Naive : 1 | Infected : 150 | Recovered : 544 | Vaccinated : 55 | Dead : 250
Day : 396 | Naive : 1 | Infected : 152 | Recovered : 540 | Vaccinated : 57 | Dead : 250
Day : 397 | Naive : 1 | Infected : 165 | Recovered : 526 | Vaccinated : 58 | Dead : 250
Day : 398 | Naive : 1 | Infected : 173 | Recovered : 517 | Vaccinated : 59 | Dead : 250
Day : 399 | Naive : 1 | Infected : 162 | Recovered : 524 | Vaccinated : 63 | Dead : 250
Mutation 17 : 4231111111 is Variant : false
```


Viral Evolution Simulation

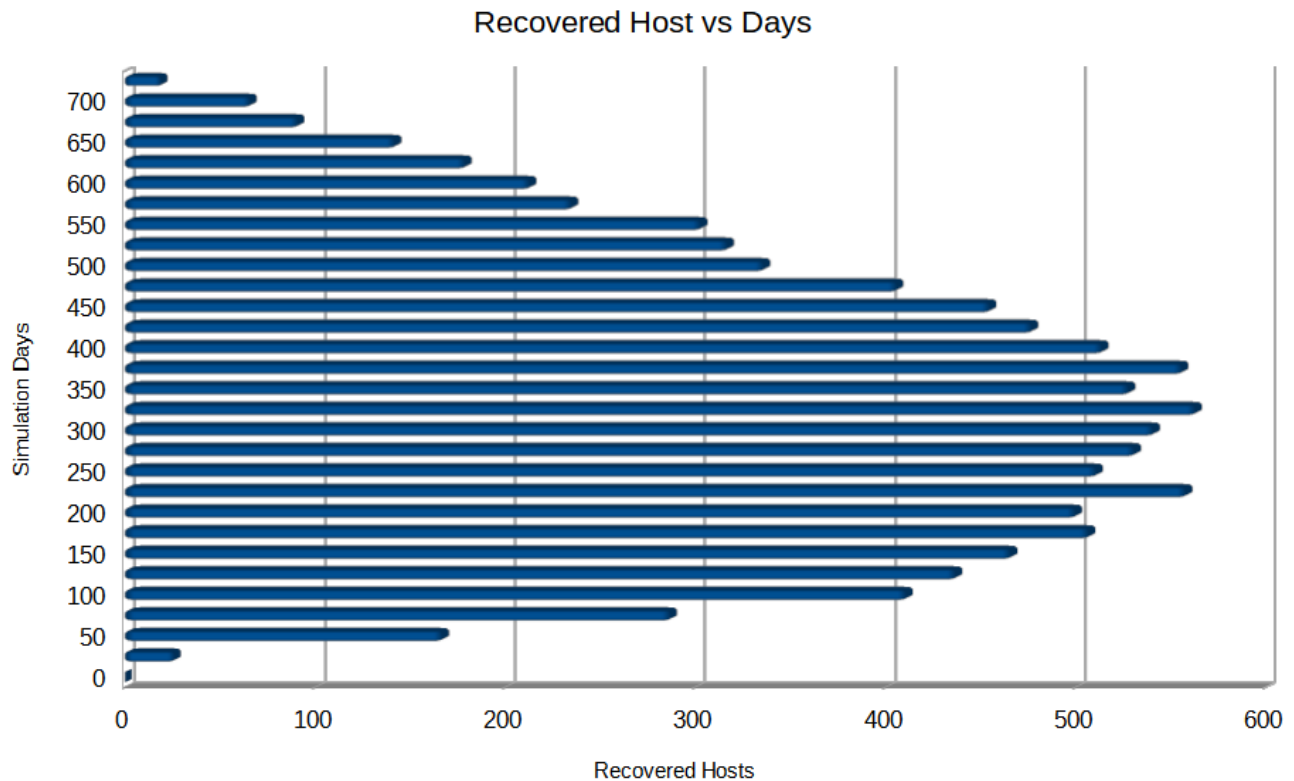
- In the above images of console outputs, observe the values of “Variant : ” for each Mutation 15, Mutation 16 and Mutation 17.
- Every “Variant : True” indicates that the fitness value of the virus inside a host is higher than the threshold and the dominant factor.

Results & Mathematical Analysis:



- It can be observed in the above graph that with our approach and logic, the simulation algorithm however randomized it may be, shows declining infection rate for the mutant generations of the virus.
- On closer observation it is also evident that although a dominant variant is arriving even after the start of the vaccination, the infection rate continues to drop.

Viral Evolution Simulation



- The above Recovered Hosts vs Days graph shows a sudden peak and then steep drop in the recovered hosts. Considering the fact that the infection rate also drops as number of days in the simulation pass. It is evident that both the infected and recovered values peak before 365 days of simulation indicating that once vaccination starts the hosts build immunity.
- Although we believe that even after being vaccinated some of the hosts still get infected because of the dominant variant of the virus or maybe because of the susceptible genome of the host.

Test Case Results:

The screenshot shows an IDE's Run window with the following details:

- Run:** All in Viral_Evolution_PSA
- Tests passed:** 14 of 14 tests – 1 s 255 ms
- Test Results Table:**

Test Case	Duration
<default package>	1 s 255 ms
RandomWalkTest	4 ms
PersonTest	861 ms
getCurrentNonInfectedListTest	478 ms
getCurrentInfectedListTest	172 ms
getPersonListTest	211 ms
SimulatorTest	364 ms
pushValueToHashTableTest	359 ms
loadHashTableTest	5 ms
PositionTest	
MutationTest	26 ms
IslandTest	
testIslandSetCenter	
testIslandGetCenter	
- Output:** /Users/sayalimahajan/Library/Java/JavaVirtualMach: mutationColor {2=java.awt.Color[r=102,g=186,b=84]}
Process finished with exit code 0
- Bottom Bar:** Git, Find, Run, TODO, Problems, Debug, Terminal, Build

Conclusion:

- After brainstorming over the entire projects problem statement we realized that representation of the gene structure is way more important than understanding the protein structures to design the algorithm for any gene evolution.
- Once the hosts are vaccinated the impact of the virus drops drastically. Proportional changes are observed in both infected and recovering graphs of the hosts are proof of that.
- With social structures and respective restrictions in place, we can use graphs and trees data structures to better understand and map down the pandemic spread.
- The virus reproduction is a shuffling method, which if used would have increased the memory consumption as well as execution time. We rather chose to ignore that and use weights of codon bases and their equal distribution in the SARS-CoV-2 gene structure.
- It can be said that in the real world virus will continue to mutate irrespective of vaccinations since we can not entirely program and build the different human genomes in algorithms to understand how the virus genotypes interact with certain humans DNA and mutate to form a dominant variant like Lambda.

References:

- https://en.wikipedia.org/wiki/Positive-sense_single-stranded_RNA_virus
- https://en.wikipedia.org/wiki/DNA_and_RNA_codon_tables#Standard_RNA_codon_table
- https://en.wikipedia.org/wiki/Genetic_code
- <https://nextstrain.org/ncov/gisaid/global>
- https://genome.ucsc.edu/cgi-bin/hgTracks?db=wuhCor1&lastVirtModeType=default&lastVirtModeExtraState=&virtModeType=default&virtMode=0&nonVirtPosition=&position=NC_045512v2%3A1%2D29903&hgside=1133869119_bR5bzYzNJAGr0pkslyjYXazsVtAZ

Video Link:

https://northeastern-my.sharepoint.com/:v:/g/personal/dasgupta_m_northeastern_edu/ERX3gTGHWt9EoClf_VGk2XUBPqosRPzcV8lq4osB1jfJHg?e=CEX69m