Cairo University
Faculty of Computers and Artificial intelligence
Department of Bioinformatics

Gene Mutation on COVID

Supervised by

Dr. Amin Allam
Dr. Ahmed Farouk
TA. HODA Ahmed

Implemented by:

Student id	Student name
20188008	**Andrew Medhat Kamal**
20188037	**Omar Ahmed Sayed**
20188053	**Nada yassen Abdelrahman**
20188025	**Sara Mostafa Ahmed**
20188039	**Lina Bassel Mohamed**

Graduation Project

Academic Year 2021/2022

Final Documentation

Table of Contents

Chapter One

Chapter Two

Chapter Three

Chapter Four

Chapter Five

List of Figures

List of Abbreviations

RBD ... Receptor Binding Domain

ACE2 ... Angiotensin-converting enzyme

NCBI … National Center for Biotechnology Information

RMSD … Root Mean Square Deviation

DNA … Deoxyribonucleic Acid

RNN … Recurrent Neural Network

PDB … Protein Data Bank

Chapter One

Introduction

1.1 Motivation

1.2 Problem definition

1.3 Project objective

1.4 Gantt chart of project time plan

1.5 Project development methodology

1.6 The used tools in the project (SW and HW)

1.7 Report Organization

1.1 Motivation

We must return to a state of stability in our lives and live them as we did before Corona arrived, which necessitates having all the available information and appropriate drugs. The newly discovered COVID-19 variants have the potential to spread quickly and cause more serious sickness than the original virus. Instead of discovering the variant after people become infected with it, contribute to vaccine development research. Protecting research from potential harm in laboratories while employing genome sequencing to find these mutations Because it does not necessitate as many human resources as genomic laboratories, the project will not be hampered by a lack of experience.

COVID-19 has the potential to be ten times more lethal than seasonal flu, which kills about 0.1 percent of those infected.COVID-19 We contemplate continued isolation and resampling many days later in this situation. Private labs' sensitivity may differ from one another. The most sensitive test is the PCR. Actually, the testing procedure must be completed in a sophisticated laboratory facility. As a result, the turnaround time for these tests can range from a few days to several weeks. And, if there's a pandemic, because there are so many samples, it will take several days longer than we would want or hope, and because of its technical limitations, this test has limited utility. So we considered our project concept. We can collect and analyses all possible mutations using this method. Whether or not they attach to human receptors Using our machine learning model to determine if it is tied to a human receptor or not, researchers can use it. Use this model to recommend potential medications. We will use a protein docking tool to ensure that the RNN model predicts correctly by RMSD and docking score. Covid-19 mutations interact with human receptors and cause harm to humans.

1.2 Problem definition

In an ideal world, we'd have an issue where corona is threatening everyone's lives, and there's no way to stop something that is a new sickness that circles incredibly quickly, much like the omicron.

Corona was unique in that it arrived rapidly and unexpectedly, and it was unaffected by a commonly used drug, it is normal for a virus to change (mutate) causing variants these variants are now spreading and can be more dangerous and more deadly than original virus. And it was tough to find a cure for it, as we struggled for a year to find a solution that would have been a lifeline for the entire country, regardless of the fact that it was not the final solution. However, the Corona also shocked us with a new mutant, and it was very costly to figure out which mutant this mutant belonged to and which solution was closest to it among the Covid solutions. And this is what we want to achieve in the world, so that if a COVID is targeted quickly, it can be determined to which family it belongs and what the nearest solution is at the time. This will bring a great deal of stability to our planet and save everyone a great deal of time and money.

Millions of people have been impacted by the COVID-19 outbreak around the world. In addition to having terrible health effects, the pandemic has crushed our objectives, upended our family dynamics and employment positions, and destroyed our economic stability. As a result, the pandemic's unparalleled worldwide crisis has had a significant influence on our mental health.

IS Global is a participant in COVID Stress, a global study that began with COVID-19's inaugural wave. Over 173,000 people from 48 nations have so far shared their stories about the pandemic. The initial poll found that being a woman, being younger, being unmarried, having a lower level of education, and caring for more children were all linked to higher levels of stress. And being a resident of a country or region where the COVID-19 situation is worse.

These findings emphasize the necessity of public health measures aimed at assisting the most vulnerable populations, who are more prone to greater levels of stress. The study also looked at how diverse psychological and behavioral reactions influenced compliance with government-imposed coronavirus-control measures. During the first wave, the data revealed significant variances between countries. In comparison to participants from other regions of the world, participants from Western European countries were more concerned about COVID-19, more anxious, and had less faith in government initiatives. The less stressed people felt, the more confident they were in their government's attempts to curb the spread of the coronavirus.

All of this demonstrates the influence of the Corona virus on our lives, as we are dealing with a novel sickness for which there is no information or medicine, and which mutates rapidly and considerably, necessitating prompt response.

Doing genome sequencing to covid-19 to identify its sequence in laboratories take a long time and many resources So what were the options for dealing with this new situation that stays up with the rest of the world while wasting a great deal of time and resources?

The solution was Genome sequencing in laboratories which involves taking a sample from a positive PCR test and amplify it, this solution take huge cost and time and has many disadvantages as The role of most of the genes in the human genome is still unknown or incompletely understood. Therefore, a lot of the “information” found in a human genome sequence is unusable at present and also Reports indicate that accuracy of RT-PCR results rely heavily on sample collection timing, type, storage, handling, and processing. The tests diagnose active infection only; they can’t detect whether an individual was infected previously. A false negative result is possible if the sample isn’t properly obtained or if an individual is tested too early after exposure to the virus or too late in their infection. So the idea is to find all conceivable COVID mutations and connect them to the common types so that researchers can recommend potential pharmaceuticals and treatment approaches much more quickly and for free, because we'll be able to construct a model that works on any COVID mutation.

If there is a RNN model that can determine whether or not a mutant is successful, and then return to the origin of the mutation to determine the likely treatment, we can even provide the mutant to COVID. We are confronted with a new therapy, such as the Omicron, which is the most similar to it, and we struggle to restore the world's greatest security in the face of new mutants.

1.3 Project objective

Our suggested solution is to build RNN model which help in predict that new sequences infect human cell (can bind using RBD and ACE2 receptor or not).

Main area of project is finding several new variants of covid-19 using which is Severe acute respiratory syndrome coronavirus 2 which known now as SARS-CoV- 2 ,is a positive RNA coronavirus is contagious between humans and is the cause COVID-19.

The diameter of the SARS-COV- 2 is about 50-200 nanometers. Like all other coronaviruses, it is composed of four structural proteins one of them called spike protein, and researcher proved that “spike” is responsible for human infection with virus by binding to the ACE2 receptor of the human cell , we are interested in the spike (S) protein, once the virus interacts with the host cell, extensive structural rearrangement of the S protein occurs, allowing the virus to fuse with the host cell membrane. The spikes are coated with polysaccharide molecules to camouflage them, evading surveillance of the host immune system during entry.

The total length of Spike protein is 1273 amino acid, it is composed of 2 subunits (s1 and s2). the S1 subunit (14–685 residues), and the S2 subunit (686–1273 residues), they are responsible for receptor binding and membrane fusion.

In s1 subunit there is an N-terminal domain (14–305 residues) and a receptor-binding domain (RBD, 319–541 residues) that recognizes and binds to human cell through a receptor called ACE2 receptor which is a protein on the surface of many human cells, s2 subunits promotes the fusion between human cell and RBD in s1 subunit.

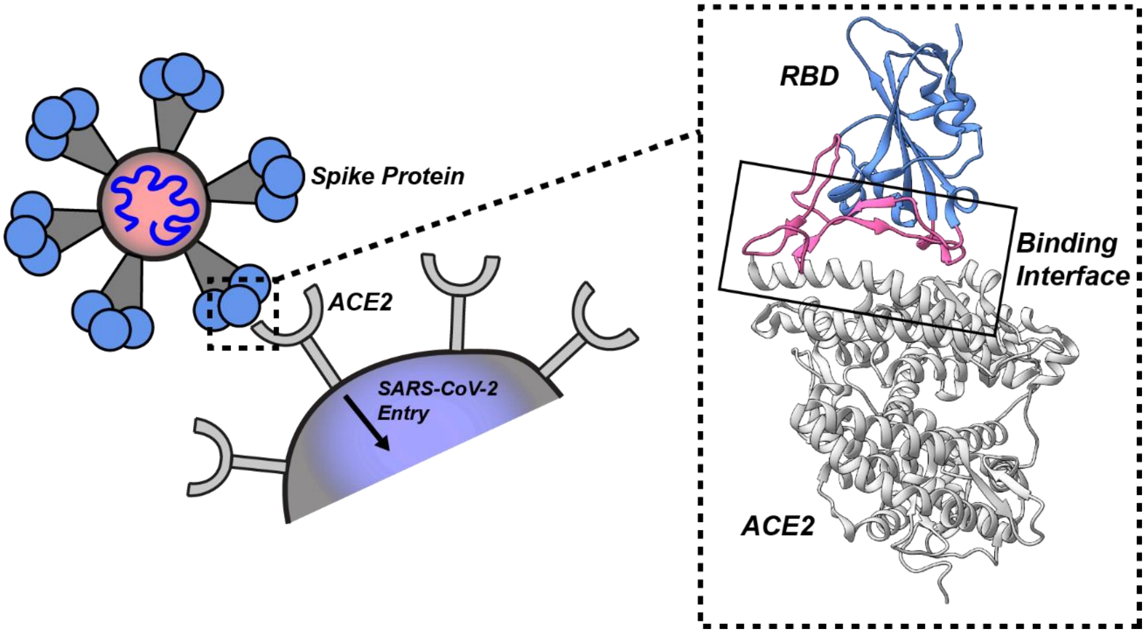
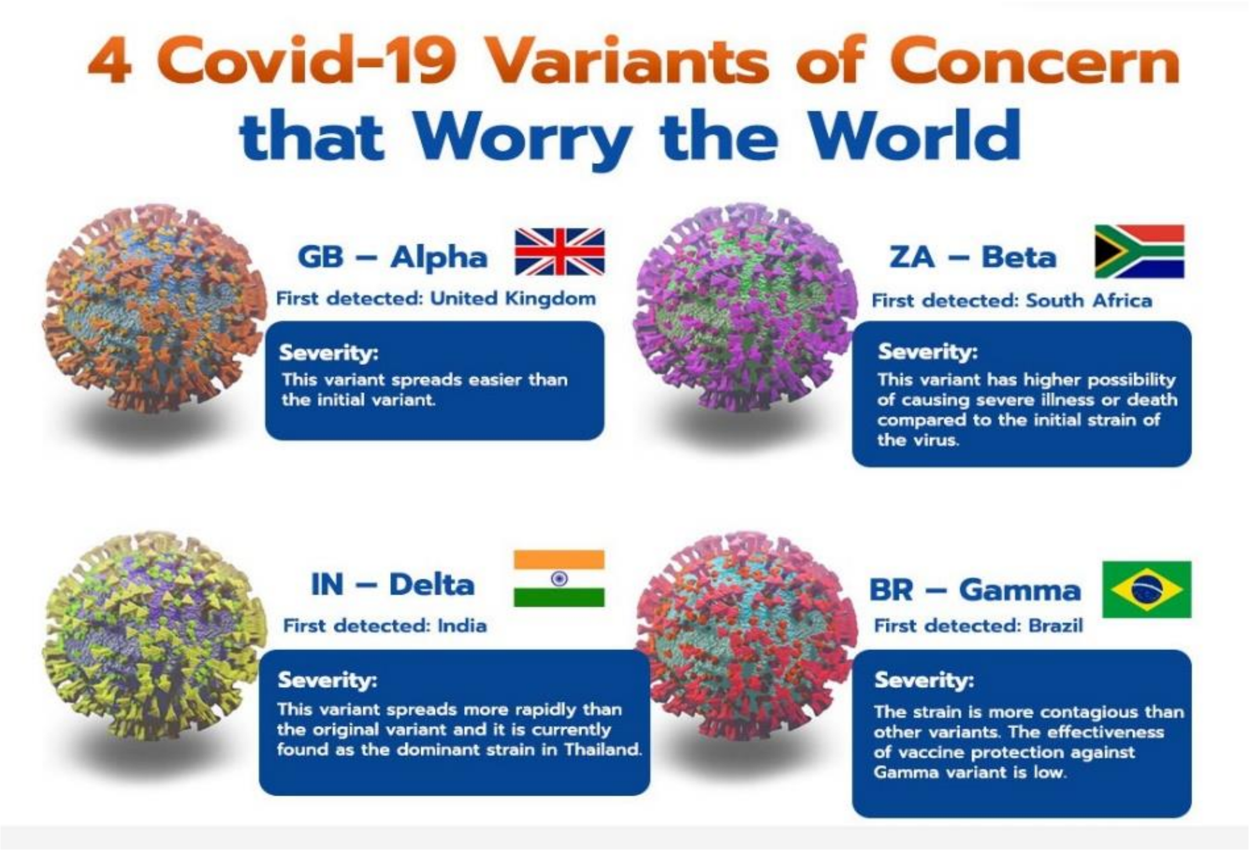


Figure 1

This figure shows binding between human cell (ACE2 receptor) and RBD of covid-19.

When happen change in amino acid in spike protein know as a mutation and the virus will have a new structure and characteristic known as a variant, that happened in SARS-CoV- 2 and we have now many variants such as (alpha, beta, gamma and delta), And many sequences of spike protein. We will work on the effective part in spike protein called RBD to predict if we have another sequence if it causes infection or not in other word if it will bind with human cell or not.

We studied variants of concern which are known to spread easily and cause more serious illness, Variants of concern that are detected are (Alpha, Beta, Gamma and Delta).



This variant spreads easier than the initial variant.

This variant has higher possibility of causing severe illness or death compared to the initial strain of the virus.

This variant spreads more rapidly than the original variant and it is currently found as the dominant strain in Thailand.

The strain is more contagious than other variants. The effectiveness of vaccine protection against Gamma variant is low.

Figure 2

This figure shows 4 variants of COVID-19, their severity and Country where they were first detected.

So, we got the sequence of each variant from online database called NCBI through searching by the variant’s scientific name

The genbank file consists of regions of the genome of the variant with it’s name and range of each region in the genome , we used code written in python which takes a fasta file containing sequence of genome and splits it according to ranges written in the genbank file.

Then we arranged these regions in an excel sheet which contains each region and a positive or negative sign which specify whether this region bind to human cell or not and RBD in the spike protein is the only region that binds to human cell. We need to increase our positive dataset so that our model won’t be biased because of the difference between number of positive and negative regions.

There were 3 solutions to increase our positive dataset:

First:

By using NCBI, we got all mutated data in NCBI discovered until now and put this data in form of excel sheet every region with its function if it bind or not the data that got from NCBI each region is known that it is positive or negative by studying region of proteins we discover that Only RBD is positive and all other region is negative and we also removed all duplicated data according to NCBI data.

Second:

By splitting RBD of the 4 variants into segments

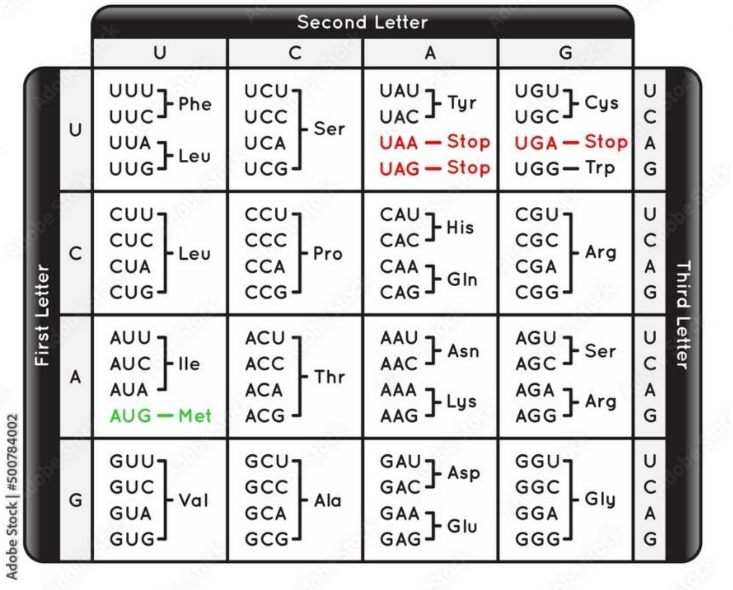
Third:

By using silent mutation:

Silent mutations occur when the change of a single DNA nucleotide within a protein-coding portion of a gene does not affect the sequence of amino acids that make up the gene's protein. they are called silent because their effect cannot be heard.

For example, AAA (codes for the amino acid lysine, Lys) being mutated to AAG (which also codes for Lys), So changing the last A to G will not affect the resulted protein of the DNA sequence

And if the protein is not affected by mutation this means that it has the same functionality, for example if we make silent mutation in RBD of one of the 4 variants it still binds to the ACE2 human receptor, also we make silent mutation to any negative region it will not bind to the ACE2 human receptor.



	Second Letter				
First Letter	U	C	A	G	Third Letter
U	UUU UUC } Phe UUA UUG } Leu	UCU UCC UCA UCG } Ser	UAU UAC } Tyr UAA — Stop UAG — Stop	UGU UGC } Cys UGA — Stop UGG — Trp	U C A G
C	CUU CUC CUA CUG } Leu	CCU CCC CCA CCG } Pro	CAU CAC } His CAA CAG } Gln	CGU CGC CGA CGG } Arg	U C A G
A	AUU AUC AUA } Ile AUG — Met	ACU ACC ACA ACG } Thr	AAU AAC } Asn AAA AAG } Lys	AGU AGC } Ser AGA AGG } Arg	U C A G
G	GUU GUC GUA GUG } Val	GCU GCC GCA GCG } Ala	GAU GAC } Asp GAA GAG } Glu	GGU GGC GGA GGG } Gly	U C A G

Adobe Stock | #500784002

Figure 3

This figure shows amino acid table

According to this table multiple codons are translated to the same amino acid like: CUU, CUC, CUA, CUG, all these codons are translated to the same amino acid which is Leu.

So we made code that changes randomly in the DNA sequence but only the changes that lead to silent mutation according to the table above, By this way we get a new mutated RBD, and it’s translated to the same protein of the original RBD, so we could increase the number of positive RBDs in the training data.

Now we will have our large space data of covid so can build our RNN model and make it learn this data to be able to detect any new mutated sequence of covid
This will help researchers determine if the current vaccines are efficient for this variant. If not, rapidly develop a new vaccine for it.

Why use RNN:

Our input is a sequence of DNA like ACTGACGTG and RNNs are a powerful and robust type of neural network due to an internal memory, RNN’s can remember important things about the input they received, Recurrent neural networks can form a much deeper understanding of a sequence and its context compared to other algorithms, it considers the current input and also what it has learned from the inputs it received previously.

which makes it perfectly suited for machine learning problems that involve sequential data.

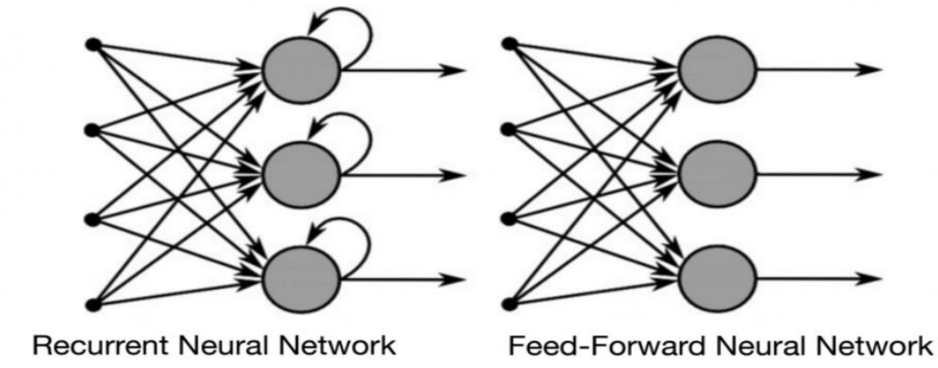


Figure 4

This figure shows RNN model vs Feed-Forward Network

Now model need to be tested to check its accuracy to be sure of it to launch it to use, so we made test data also by using silent mutation that we mentioned above.

Then we used Protein docking tool to guide us:

First, we used Swiss model tool:

It is a server for automated comparative modeling of three-dimensional (3D) protein structures.

Comparative modelling: also known as Homology modeling: is modelling of a protein structure using a known experimental structure of a homologous protein as a template, as homologous proteins have similar 3D structures and can be used as templates for modelling.

Building a homology model includes 4 steps which are:

1) Determining a structural template.

2) Alignment between the target sequence and the template structure which is determining the regions of similarity between the 2 sequences.

3) Model building

4) Model quality evaluation.

It takes an amino acid sequence of a protein as an input to build a 3D model for it.

3D structure of a protein:

Protein is a sequence of amino acids, that is made up of a basic amino group ($-NH_2$), an acidic carboxyl group ($-COOH$), and an organic R group (or side chain) that is unique to each amino acid , when a peptide bond links carboxyl group of one amino acid to the amine group of the next amino acid it forms unbranched chain of amino acids and 3D structure of a protein is the level of protein structure at which an entire polypeptide chain has folded into a three-dimensional structure which is useful in determining the protein function and information about protein-protein interaction.

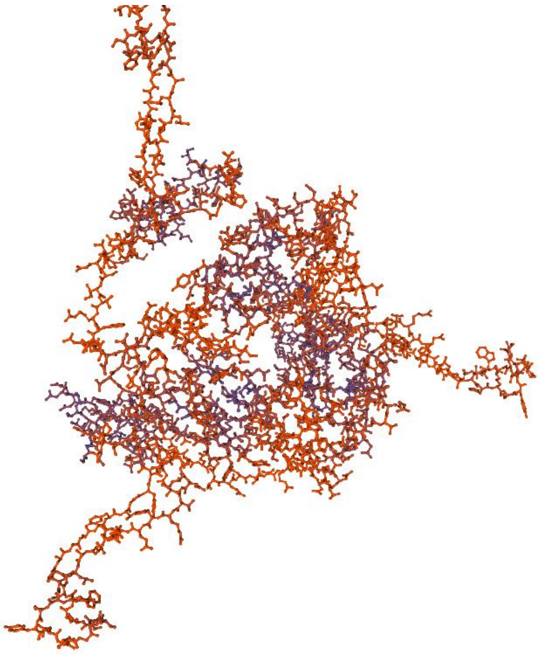


Figure 5

This figure shows the 3D structure of receptor Alpha RBD

Sometimes this tool builds more than one model so we choose the one with higher sequence identity percentage which is sequence similarity between the target and the template sequence and sometimes the sequence identity is low (below 40%) for this reason we used a translation tool called “expasy” which takes DNA sequence as an input and outputs 6 frames of the resulted protein.

For ex:if we give the sequence "ATGCGATCGGACAGTCGAGTCCAGTAGACGATC" to expasy

M R S D S R V Q - T I   will be the first reading frame.

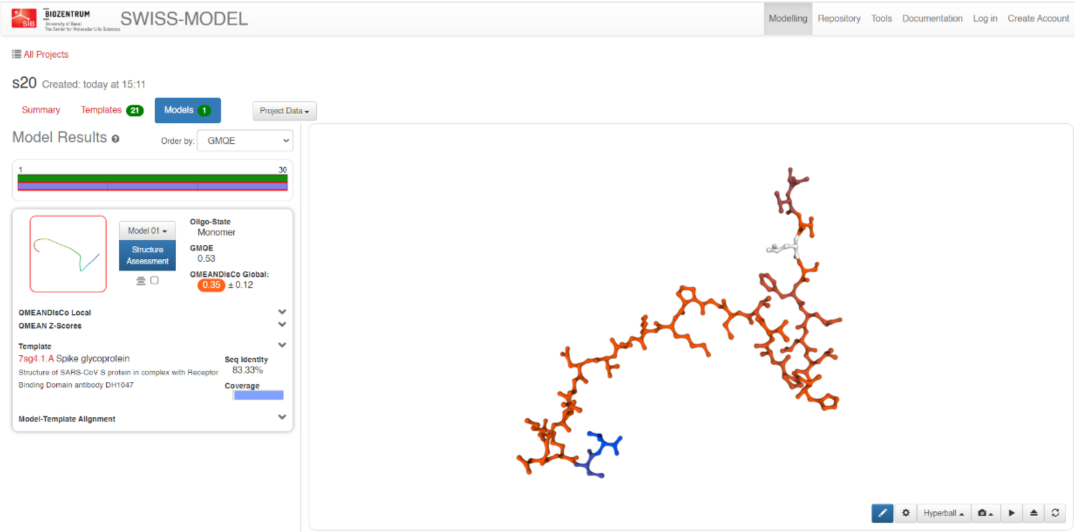
C D R T V E S S R R   will be the second reading frame.

A I G Q S S P V D D   will be the third reading frame.

The first reading frame starts with a Methionine (M) encoded by the ATG codon but if we were to consider the second reading frame and therefore to start "reading" the code from the second base of the nucleotide sequence, the first amino acid to be read would be (C) encoded by the TGC codon. Moreover, if we didn't know the orientation of the nucleotide sequence, the translation could be read either in the forward (5'->3') or the reverse (3'->5') giving an additional three possible ways of reading the code.

Such as ORF3A of beta variant, when we use the first frame of translation, the models that appear in Swiss model have maximum sequence identity of 32.14% but when we use the third frame, we get models with sequence identity 99%.

So, we tried more than one frame to get the best possible sequence identity.



s20 Created: today at 15:11

Model Results

Order by: GMQE

1 30

Oligo-State
Monomer

GMQE
0.53

QMEANDisCo Global:
0.35 ± 0.12

QMEANDisCo Local

QMEAN Z-Scores

Template
7sg4.1.A Spike glycoprotein
Structure of SARS-CoV S protein in complex with Receptor Binding Domain antibody DH1047

Seq Identity
83.33%

Coverage

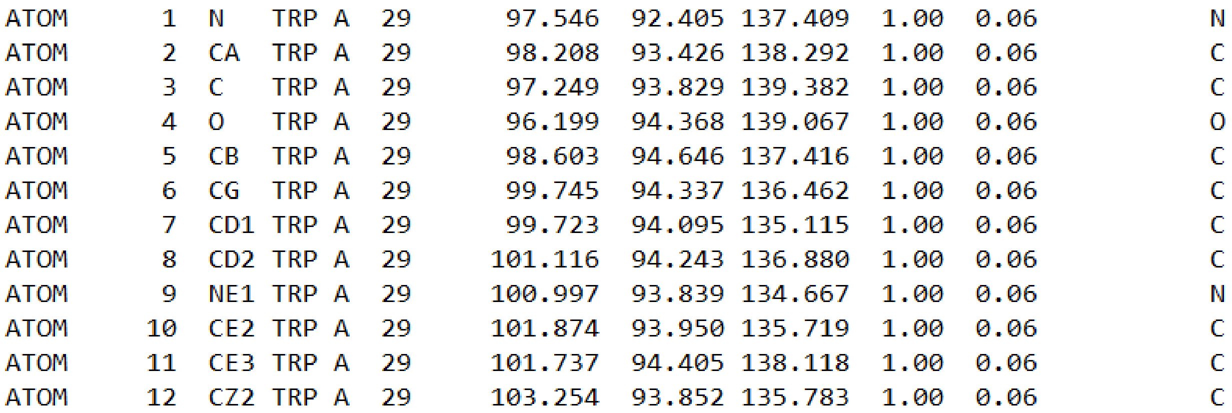
Model-Template Alignment

Figure 6

This figure shows results of Swiss model tool.

Then we download the PDB file format of the protein sequence which is the primary information stored in the PDB archive which consists of coordinate files for biological molecules. These files list the atoms in each protein, and their 3D location in space.

It consists of the number of atom , the type of the atom , the type of amino acid by it’s 3 letter character , what chain it is part of , the number of amino acid in the sequence and finally the x y z coordinates.

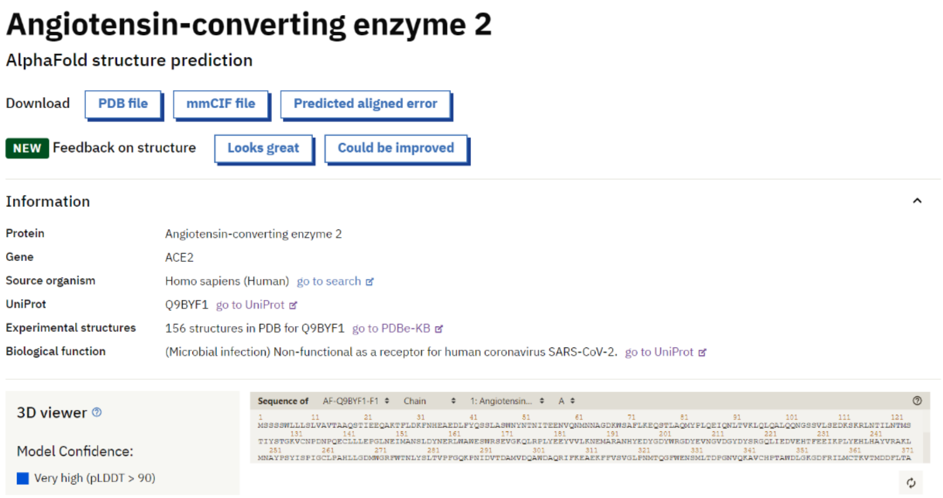


```
ATOM      1  N   TRP A  29      97.546  92.405 137.409  1.00  0.06           N
ATOM      2  CA  TRP A  29      98.208  93.426 138.292  1.00  0.06           C
ATOM      3  C   TRP A  29      97.249  93.829 139.382  1.00  0.06           C
ATOM      4  O   TRP A  29      96.199  94.368 139.067  1.00  0.06           O
ATOM      5  CB  TRP A  29      98.603  94.646 137.416  1.00  0.06           C
ATOM      6  CG  TRP A  29      99.745  94.337 136.462  1.00  0.06           C
ATOM      7  CD1 TRP A  29      99.723  94.095 135.115  1.00  0.06           C
ATOM      8  CD2 TRP A  29     101.116  94.243 136.880  1.00  0.06           C
ATOM      9  NE1 TRP A  29     100.997  93.839 134.667  1.00  0.06           N
ATOM     10  CE2 TRP A  29     101.874  93.950 135.719  1.00  0.06           C
ATOM     11  CE3 TRP A  29     101.737  94.405 138.118  1.00  0.06           C
ATOM     12  CZ2 TRP A  29     103.254  93.852 135.783  1.00  0.06           C
```

Figure 7

This figure shows PDB file

Then we searched for ACE2 human receptor in Uniprot and we got it’s accession number which is Q9BYF1, then we searched by this code in AlphaFold which is (an artificial intelligence (AI) program which performs predictions of protein structure.)



Angiotensin-converting enzyme 2

AlphaFold structure prediction

Information

Protein	Angiotensin-converting enzyme 2
Gene	ACE2
Source organism	Homo sapiens (Human) go to search
UniProt	Q9BYF1 go to UniProt
Experimental structures	156 structures in PDB for Q9BYF1 go to PDBe-KB
Biological function	(Microbial infection) Non-functional as a receptor for human coronavirus SARS-CoV-2. go to UniProt

3D viewer

Model Confidence:

🟦 Very high (pLDDT > 90)

Sequence of AF-Q9BYF1-F1 Chain 1: Angiotensin... A

```
1         11        21        31        41        51        61        71        81        91        101       111       121
MSSSSWLLLSLVAVTAAQSTIEEQAKTFLDKFNHEAEDLFYQSSLASWNYNTNITEENVQNMNNAGDKWSAFLKEQSTLAQMYPLQEIQNLTVKLQLQALQQNGSSVLSEDKSKRLNTILNTMS
    131       141       151       161       171       181       191       201       211       221       231       241
TIYSTGKVCNPDNPQECLLLEPGLNEIMANSLDYNERLWAWESWRSEVGKQLRPLYEEYVVLKNEMARANHYEDYGDYWRGDYEVNGVDGYDYSRGQLIEDVEHTFEEIKPLYEHLHAYVRAKL
 251       261       271       281       291       301       311       321       331       341       351       361       371
MNAYPSYISPIGCLPAHLLGDMWGRFWTNLYSLTVPFGQKPNIDVTDAMVDQAWDAQRIFKEAEKFFVSVGLPNMTQGFWENSMLTDPGNVQKAVCHPTAWDLGKGDFRILMCTKVTMDDFLTA
```

Figure 8

This figure shows ACE2 human receptor in AlphaFold tool

Then we downloaded the PDB file of the human receptor.

DOCKING:

It is the process of predicting how small molecules bind to a receptor of known 3D structure.

Sampling and scoring are done during docking, docking attempts to sample every binding mode between any two different structures that it is given. Then, during and/or after sampling, the sampled binding modes are ranked using a score algorithm.

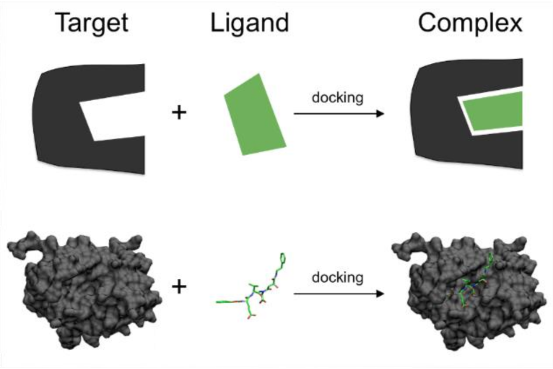


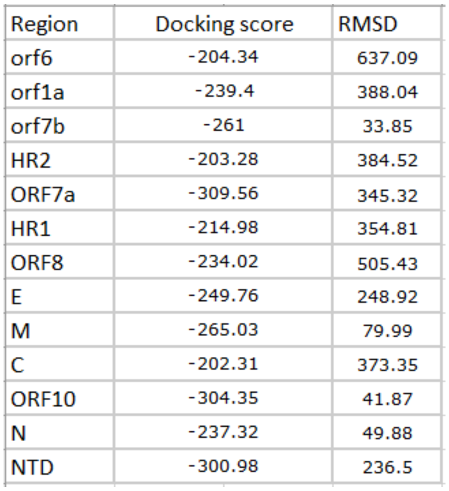
Figure 9

This figure shows the docking between the target(receptor) and the ligand.

We chose Hdock tool to perform docking between ace2 human receptor and various mutated RBDs of the 4 types of coronavirus which are Alpha, Beta, Gamma and Delta, and also to perform docking between ace2 human receptor and negative and positive regions of coronavirus.

Hdock tool takes 2 inputs (receptor and ligand) we entered them as PDB files.

First, we entered ACE2 human receptor in the receptor field and negative regions in the ligand field and we saved the docking score and RMSD in excel sheet.

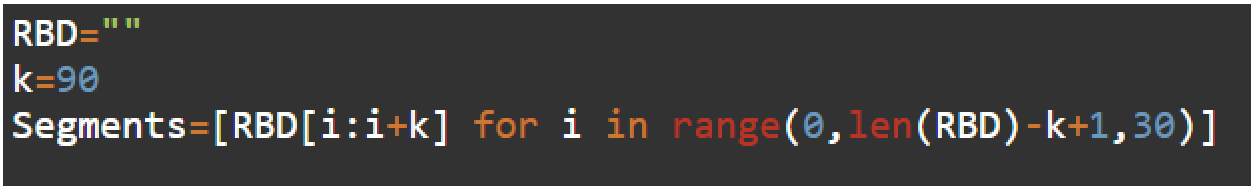


Region	Docking score	RMSD
orf6	-204.34	637.09
orf1a	-239.4	388.04
orf7b	-261	33.85
HR2	-203.28	384.52
ORF7a	-309.56	345.32
HR1	-214.98	354.81
ORF8	-234.02	505.43
E	-249.76	248.92
M	-265.03	79.99
C	-202.31	373.35
ORF10	-304.35	41.87
N	-237.32	49.88
NTD	-300.98	236.5

Figure 10

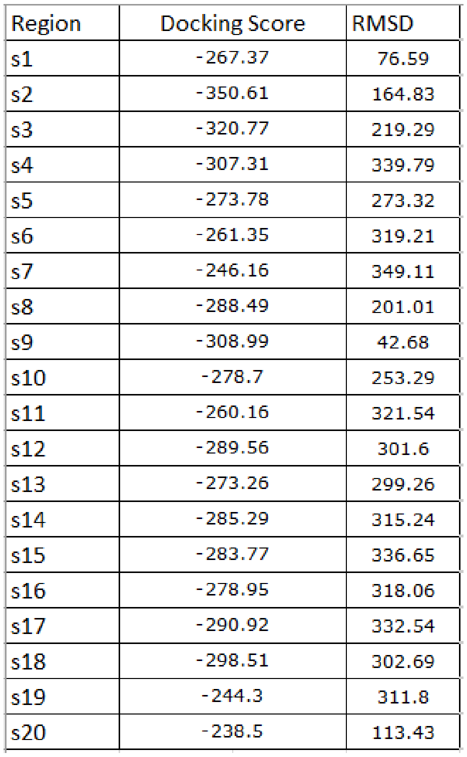
This figure shows the docking result between ACE2 and negative regions

And due to the problem of having just 4 RBD regions we divided RBD of each variant into segments with overlap through this code:



```
RBD=""
k=90
Segments=[RBD[i:i+k] for i in range(0,len(RBD)-k+1,30)]
```

This results of 20 segments each of length 90 characters with overlap of 30 characters between all segments and similar to negative regions, we entered ace2 human receptor in the receptor field and these segments in the ligand field and we saved the docking score and RMSD in excel sheet.



Region	Docking Score	RMSD
s1	-267.37	76.59
s2	-350.61	164.83
s3	-320.77	219.29
s4	-307.31	339.79
s5	-273.78	273.32
s6	-261.35	319.21
s7	-246.16	349.11
s8	-288.49	201.01
s9	-308.99	42.68
s10	-278.7	253.29
s11	-260.16	321.54
s12	-289.56	301.6
s13	-273.26	299.26
s14	-285.29	315.24
s15	-283.77	336.65
s16	-278.95	318.06
s17	-290.92	332.54
s18	-298.51	302.69
s19	-244.3	311.8
s20	-238.5	113.43

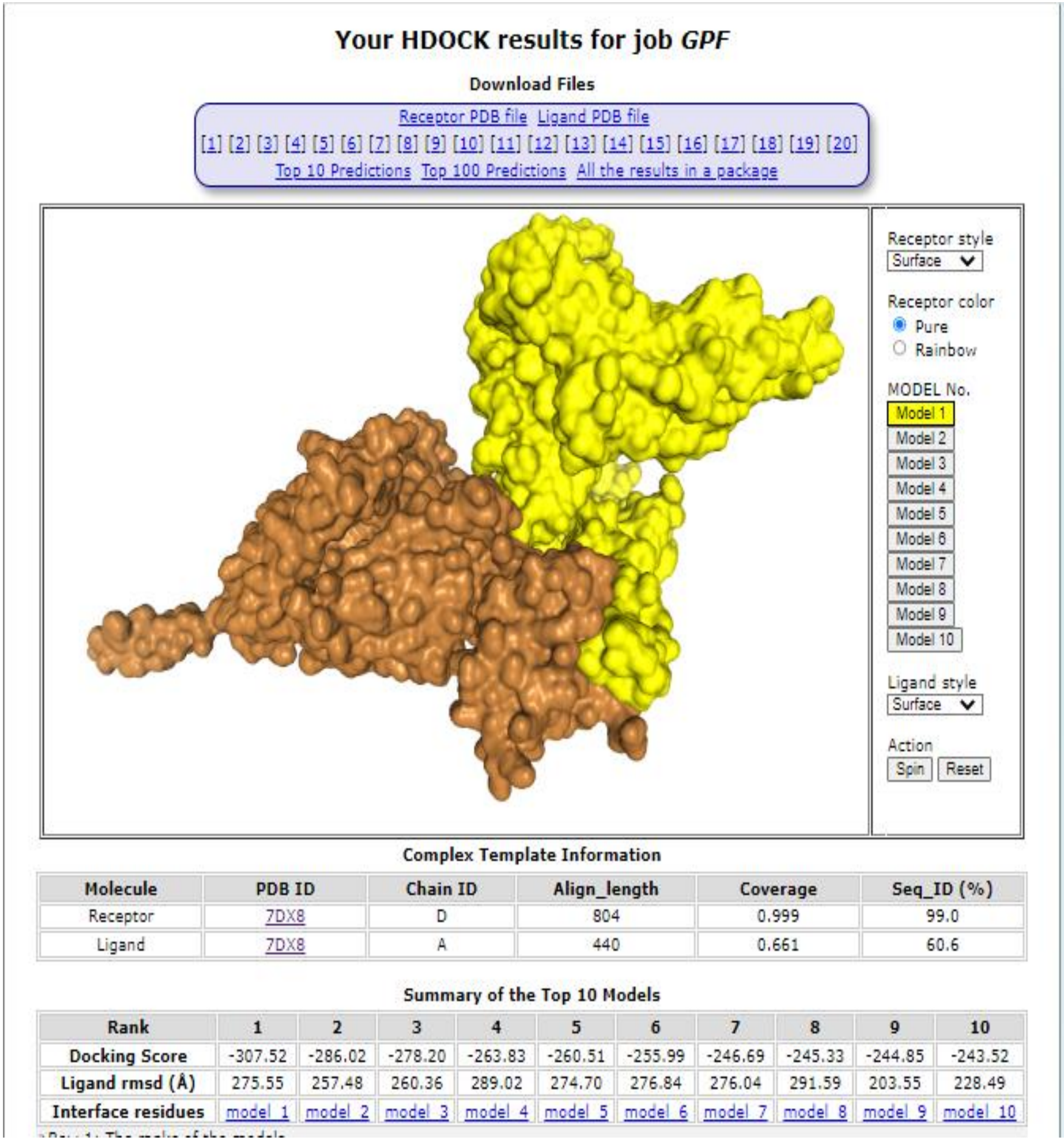
Figure 11

This figure shows the docking result between ACE2 and positive regions.

Where the docking score is the scoring function used to predict the binding affinity of both ligand and receptor once it is docked., **Root Mean Square Deviation (**RMSD) is the most used quantitative metric for determining how similar two atomic coordinates are. Values for RMSD are displayed in and computed using Å.

$$RMSD = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_i^2}$$

Where N is the number of atoms in the ligand, and d_i is the Euclidean distance between the ith pair of corresponding atoms.



Your HDOCK results for job *GPF*

Download Files

Receptor PDB file Ligand PDB file

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20]

Top 10 Predictions Top 100 Predictions All the results in a package

Receptor style: Surface

Receptor color
● Pure
○ Rainbow

MODEL No.

Ligand style: Surface

Action

Complex Template Information

Molecule	PDB ID	Chain ID	Align_length	Coverage	Seq_ID (%)
Receptor	7DX8	D	804	0.999	99.0
Ligand	7DX8	A	440	0.661	60.6

Summary of the Top 10 Models

Rank	1	2	3	4	5	6	7	8	9	10
Docking Score	-307.52	-286.02	-278.20	-263.83	-260.51	-255.99	-246.69	-245.33	-244.85	-243.52
Ligand rmsd (Å)	275.55	257.48	260.36	289.02	274.70	276.84	276.04	291.59	203.55	228.49
Interface residues	model 1	model 2	model 3	model 4	model 5	model 6	model 7	model 8	model 9	model 10

This figure shows the result of Docking tool

Then we tried to set a threshold for the negative and positive regions from these excel sheets to ensure that RNN model predicts the mutated sequences correctly.

We depend mainly on the RMSD, the lower RMSD value the better binding between the ligand and the receptor

After analyzing the docking score and the RMSD of negative and positive regions of the 4 variants we tried to set 2 thresholds for every variant, one for negative regions and the other for positive regions.

Thresholds are:

Gamma for negative regions: RMSD <=260 and score >=-355

Gamma for positive regions: RMSD <=320 and score >=-350

Delta for negative regions: RMSD <=160 and score >=-280

Delta for positive regions: RMSD <=295 and score >=-310

Alpha for negative regions: RMSD <=360 and score >=-310

Alpha for positive regions: RMSD <=340 and score >=-355

Beta for negative regions: RMSD <=325 and score >=-360

Beta for positive regions: RMSD <=320 and score >=-360

Then we made mutation where it is **a change in the DNA sequence of an organism**.

Mutations can result from errors in DNA replication during cell division, exposure to mutagens or a viral infection.

Types of DNA mutations are base substitutions, deletions and insertions.

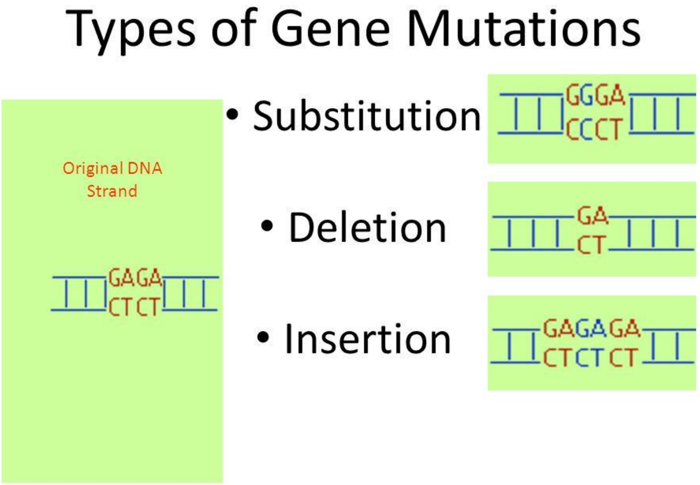


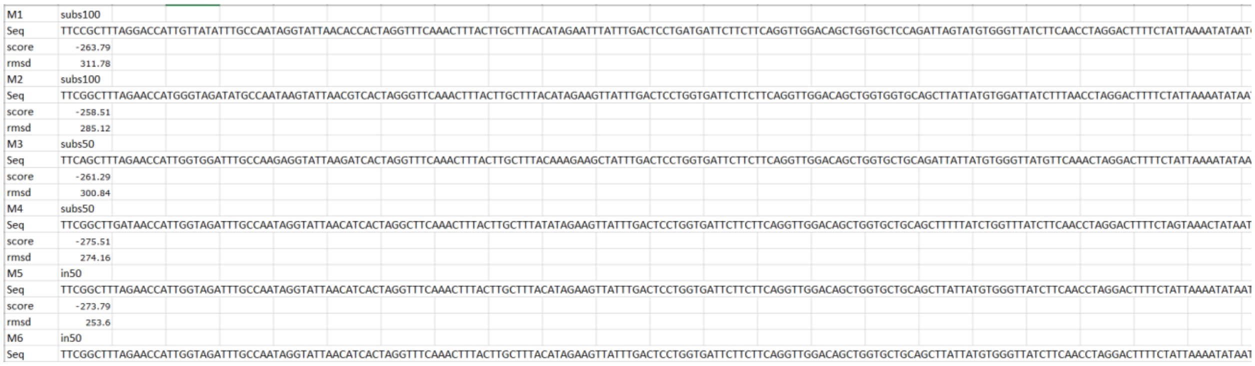
Figure 12

This figure shows different types of mutation

We made the 3 types of mutations to the RBDs of the 4 variants randomly by the following code

Which takes the RBD and a number n as parameters where n is the number of positions that will be substituted or deleted or inserted , then we get random position starting from 0 to the length of the RBD .

After making these mutations to each RBD randomly starting from 20 positions to 270 positions, we got the PDB files of these mutated RBDs through using Swiss model tool as mentioned before, then we used Hdock tool to calculate the docking score and RMSD of each mutated RBD



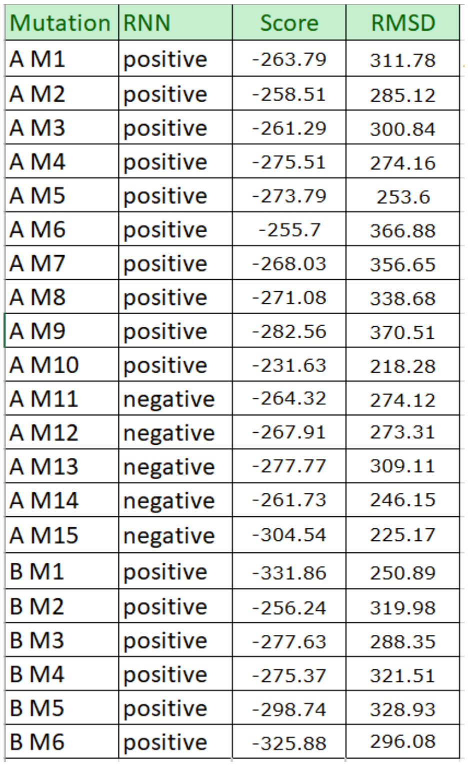
M1	subs100
Seq	TTCCGCTTTAGGACCATTGTTATATTTGCCAATAGGTATTAACACCACTAGGTTTCAAACTTTACTTGCTTTACATAGAATTTATTTGACTCCTGATGATTCTTCTTCAGGTTGGACAGCTGGTGCTCCAGATTAGTATGTGGGTTATCTTCAACCTAGGACTTTTCTATTAAAATATAAT
score	-263.79
rmsd	311.78
M2	subs100
Seq	TTCGGCTTTAGAACCATGGGTAGATATGCCAATAAGTATTAACGTCACTAGGGTTCAAACTTTACTTGCTTTACATAGAAGTTATTTGACTCCTGGTGATTCTTCTTCAGGTTGGACAGCTGGTGGTGCAGCTTATTATGTGGATTATCTTTAACCTAGGACTTTTCTATTAAAATATAA
score	-258.51
rmsd	285.12
M3	subs50
Seq	TTCAGCTTTAGAACCATTGGTGGATTTGCCAAGAGGTATTAAGATCACTAGGTTTCAAACTTTACTTGCTTTACAAAGAAGCTATTTGACTCCTGGTGATTCTTCTTCAGGTTGGACAGCTGGTGCTGCAGATTATTATGTGGGTTATGTTCAAACTAGGACTTTTCTATTAAAATATAA
score	-261.29
rmsd	300.84
M4	subs50
Seq	TTCGGCTTGATAACCATTGGTAGATTTGCCAATAGGTATTAACATCACTAGGCTTCAAACTTTACTTGCTTTATATAGAAGTTATTTGACTCCTGGTGATTCTTCTTCAGGTTGGACAGCTGGTGCTGCAGCTTTTTATCTGGTTTATCTTCAACCTAGGACTTTTCTAGTAAACTATAAT
score	-275.51
rmsd	274.16
M5	in50
Seq	TTCGGCTTTAGAACCATTGGTAGATTTGCCAATAGGTATTAACATCACTAGGTTTCAAACTTTACTTGCTTTACATAGAAGTTATTTGACTCCTGGTGATTCTTCTTCAGGTTGGACAGCTGGTGCTGCAGCTTATTATGTGGGTTATCTTCAACCTAGGACTTTTCTATTAAAATATAAT
score	-273.79
rmsd	253.6
M6	in50
Seq	TTCGGCTTTAGAACCATTGGTAGATTTGCCAATAGGTATTAACATCACTAGGTTTCAAACTTTACTTGCTTTACATAGAAGTTATTTGACTCCTGGTGATTCTTCTTCAGGTTGGACAGCTGGTGCTGCAGCTTATTATGTGGGTTATCTTCAACCTAGGACTTTTCTATTAAAATATAAT

Figure 13

This figure shows the docking result of mutated RBD with ACE2

Then we used the RNN model to know whether these mutations will bind to human receptor or not (positive or negative)

And we created excel sheet which contains the Result of the RNN model and the docking score and RMSD from Hdock tool of the mutated RBDs.



Mutation	RNN	Score	RMSD
A M1	positive	-263.79	311.78
A M2	positive	-258.51	285.12
A M3	positive	-261.29	300.84
A M4	positive	-275.51	274.16
A M5	positive	-273.79	253.6
A M6	positive	-255.7	366.88
A M7	positive	-268.03	356.65
A M8	positive	-271.08	338.68
A M9	positive	-282.56	370.51
A M10	positive	-231.63	218.28
A M11	negative	-264.32	274.12
A M12	negative	-267.91	273.31
A M13	negative	-277.77	309.11
A M14	negative	-261.73	246.15
A M15	negative	-304.54	225.17
B M1	positive	-331.86	250.89
B M2	positive	-256.24	319.98
B M3	positive	-277.63	288.35
B M4	positive	-275.37	321.51
B M5	positive	-298.74	328.93
B M6	positive	-325.88	296.08

Figure 14

This figure shows RNN result and Docking tool result of mutated RBD.

After that we check for the result of the RNN model using HDOCK result by the thresholds we set to each variant and we realized that HDOCK can guide us by 85%.

Finally Make interface to link our model with that website that help User Scientist or researcher use our model easily by Just entering RBD region then our model will act as helpful tool to detect this RBD of mutated covid sequence it can infect human cell or not this interface made by Css3, Html5 and JS to create the design of website the use flask library in python to connect between the interface and our helpful model also

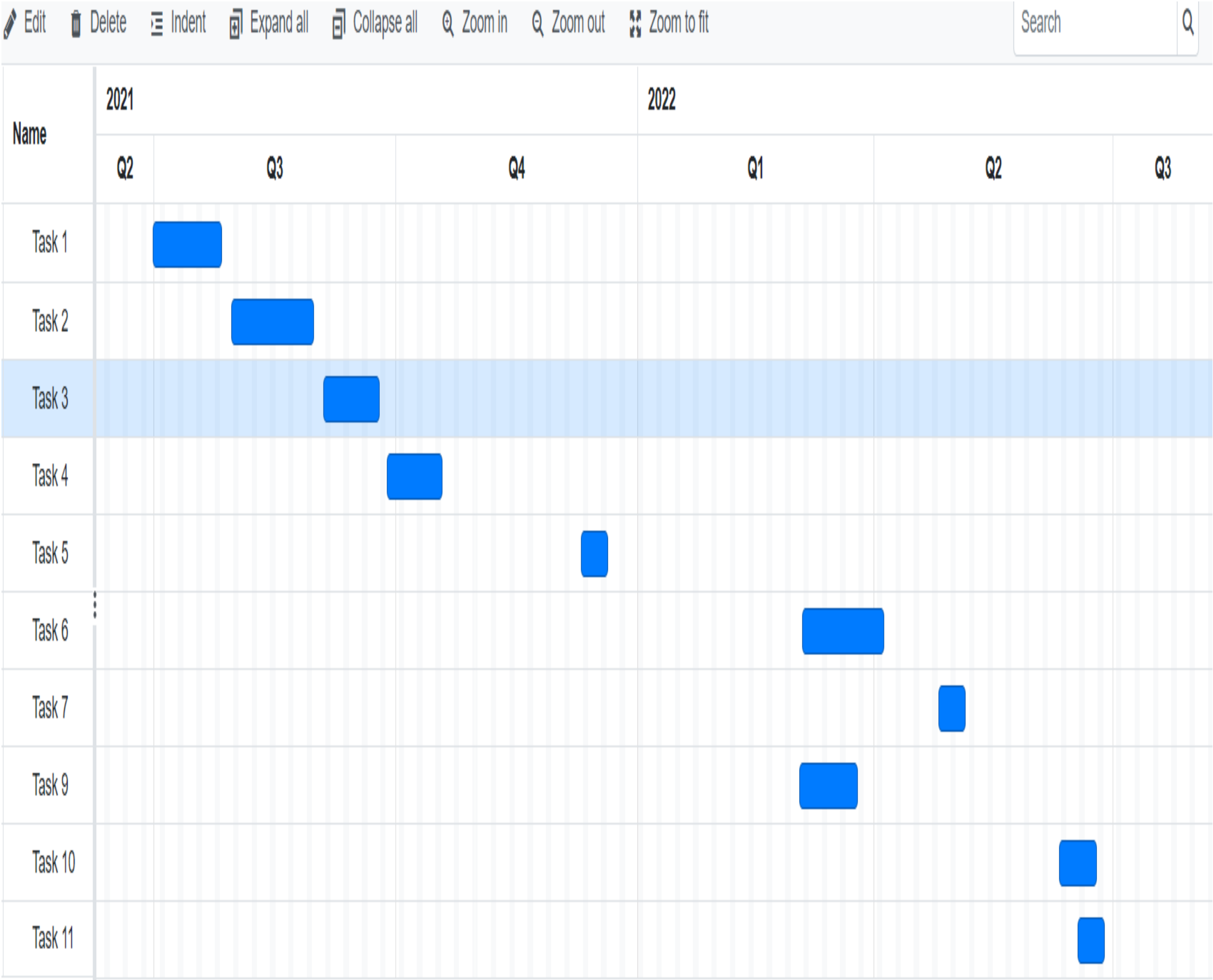
Using flask because:

Flask is a lightweight WSGI web application framework**.** Itis designed to make getting started quick and easy**,** with the ability to scale up to complex applications. It began as a simple wrapper and has become one of the most popular Python web application frameworks.

Figure 15

This figure shows building web front-end for python with flask

1.4 Gantt chart of project time plan



Gantt chart analysis:

Task	Date	Describe
1	30/6/2021 - 24/7/2021	Project idea
2	30/7/2021 - 28/8/2021	Collect variants
3	3/9/2021 - 24/9/2021	Data analysis
4	25/9/2021 -20/10/2021	ACE2 detection
5	10/12/2021 -27/12/2021	Study classifier
6	4/3/2022 - 9/4/2022	Build classifier
7	24/4/2022- 5/5/2022	Collect NCBI data
8	4/6/2022- 18/6/2022	Model adjustment
9	3/3/2022 - 25/3/2022	Mutate data
10	10/6/2022- 24/6/2022	Testing
11	18/6/2022 - 25/6/2022	Project interface

1.5 Project development methodology

In order to achieve objective:

Get 4 main variants from NCBI

Get all detected variants from NCBI

Make mutation to increase training data

Remove duplicates to avoid overfitting of model

Study classifiers and choose RNN to be our classifier

Build RNN model

Make mutation to create our test data

Test model and make high accuracy

Use protein docking tool to guide us by 85%

Make interface to easily represent our work

Connect our model with the interface to be easily connected

Now model is trained and its data is prepared and model is trained with high performance and can detect any unknown sequence.

That's our project development result take input RBD DNA sequence to predict whether that infect human cell or not by entering it to trained RNN model to predict, if positive so it means that it will infect human body and if negative that's means that it will not infect human body or harm it.

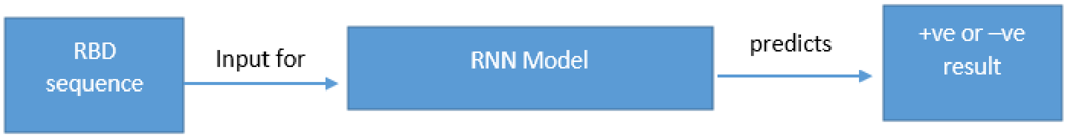


Figure 16

Project methodology

1) Input:
   DNA sequence (Only RBD region) need to be detected.
2) Process:
   Detect RBD can infect human cell or not by trained RNN model.
3) Output:
   Positive means can infect human cell.
   Negative means can't infect human cell.

1.6 The used tools in the project

We have used some tools to help us reach our objective, we don't use any hardware tools (except our computers to run software tools we will mention on it) we depended on software tools in order to produce the RNN model and New mutated data

Software tools used:

1) **Visual studio code**
   That helps in running any PY code that enables us to:
   - Put data in preferred form.
   - Make mutation.
   - Build RNN model and also save it and connect it with website (using flask).

   And also run HTML5, CSS3 and JS that used to build our interface.

2) **Expasy tool**

We used this tool to translate DNA (RBD sequence) into 6 frames of amino acid sequence(protein).

3) **Swiss Model**

We used this tool to generate PDB file of the translated protein.

4) **Hdock Server**

We used docking tool to ensure that the RNN model predicts the binding between ACE2 receptor and the mutated RBD sequence as it outputs the docking(binding) score and the RMSD which is the root mean square deviation.

5) **pdb2fasta tool**

We used this tool to convert PDB structure to FASTA sequence

6) **AlphaFold**

We used this tool to get the PDB file of ACE2 human receptor.

1.7 Report Organization

Our project about corona virus specially sars cov2 or what is called (covid 19), researchers proved that “spike” is responsible for human infection with virus by binding to the ACE2.

receptor of the human cell, so we are interested in the spike (S1) protein specially region called RBD, once the virus interacts with the

host cell, extensive structural rearrangement of the S protein occurs, allowing the virus to fuse with the host cell membrane. When there are changes in its RNA, this will lead to a mutation, and we do not know if the currently produced drug will be effective for this mutation or mutations that may occur in the future or not, and in this case we'd have an issue where corona is threatening everyone's lives, and there's no way to stop something that is a new sickness that circles incredibly quickly, much like the omicron (new mutation of SARS-COV 2).

Therefore, we proposed this project in order to predict several of the possible mutations that may occur in the future and also to predict by entering sequence if it can bind to ACE 2(human receptor) and cause this infection or not, and that is done through the use of machine learning, especially Recurrent neural network (RNN model).

we collect our training data from NCBI, and make silent mutation in this data to use for testing out model. Finally, we made several types of mutations and used the docking tool to guide us in whether these new sequences are already correctly predicted by our model or not and the success rate of the modal in predicting new mutations that were not previously recognized.

Chapter Two

Related Work

Because this was a first-time expedition, there were no previous implementations, but there was a method which is genomic sequencing. Although genome sequencing is frequently compared to "decoding," a sequence is still a code. A genomic sequence is, in some ways, just a long series of letters in a cryptic language. When you read a sentence, the meaning is not solely determined by the letter sequence. It's also in the words those letters form, as well as the language's syntax. The human genome, likewise, is more than just its sequence. Consider the genome as a book with no capitalization or punctuation, no breaks between words, sentences, or paragraphs, and a smattering of gibberish letters between and even inside phrases .By performing this technique on a patient's sample and submitting it to a PCR test to detect the virus's sequence, the virus's sequence can be determined.

Researchers used genomic sequencing in laboratories to uncover novel variants in order to find the mutation. The virus's genetic sequence was retrieved from a patient sample using a PCR test to detect and amplify the virus's genome (producing many copies), and then compared to other sequences in worldwide databases.

It’s advantages:

- Identify Viruses, understand their origin and transmission.
- Help in designing candidate vaccines.
- Help in developing new strategies to prevent spreading.

It’s Disadvantages:

- High cost, work and some techniques are too time consuming.
- Possible harm for participants.
- Participants may have little experience in sequencing of viral genome.
- Require many human resources.
- Probability of staff sickness and unavailability.
- Affected by quality of sample.

- PCR test is considered to be the gold standard for the early detection of virus, but this test has limited application to use as bedside test because of its technical complexity

Furthermore, the docking tool, which provides that and is the only accessible method for testing the binding between the human receptor and the virus but it has many disadvantages.

It’s Disadvantages:

- Docking scores are generally bad at telling which binding conformations are useful,
- It needs PDB file to be taken as an input not just DNA.
- It takes an hour or more to send the result of the docking

However, our method will save time and effort because it will only require a sequence of variants to predict whether or not they cause infection using machine learning.

As previously stated, researchers discovered novel mutations in the form of lab trials employing PCR-tests on patient samples. Unlike prior solutions, our project concept does not rely on lab experiments. It will be determined through the use of computer technology and the recurrent neural network technique, which is a machine learning algorithm. We don't need to take a patient's sample for each mutation; instead, we'll look for fasta format files for COVID-19 variations and operate on them. We'll work on Alpha, Beta, Delta, and Gamma variations, and we'll get a lot of mutations for them to train our model by take all sequences found on NCBI and make new sequences by mutation to let model learn all possible variants then when It is used by user, he can enter only RBD sequence and can say if t bind or not if it can infect or not.

Chapter Three

System analysis

3.1 Project specification

3.1.1 Functional requirements

3.1.2 Non-functional requirements

3.2 Use case diagram

3.1 Project specification

3.1.1 Functional requirements

1. **Validity checks on the inputs:**

   1)The input field accept sequences with length of 600 characters.
   2)The input field will accept only sequence DNA (Contains only A, C, G, T).

2. **Operations**

   1) When The input field entered it taken and check if spliced into word or not, if not it will be spliced using python code.
   Then entered to test it by the trained model to check if it is positive or negative effect.

3.1.2 Non-functional requirement

1-Flexibility:

Reach user’s requirements as quickly as possible as the interface provides result within 10 sec.

2-Usability:

Having friendly user interface that provides facilities to all users to detect whether the input RBD can infect human cell or not.

3-Reliability:

The tool should provide large search space of covid to be more accurate.

3.2 use case Diagram

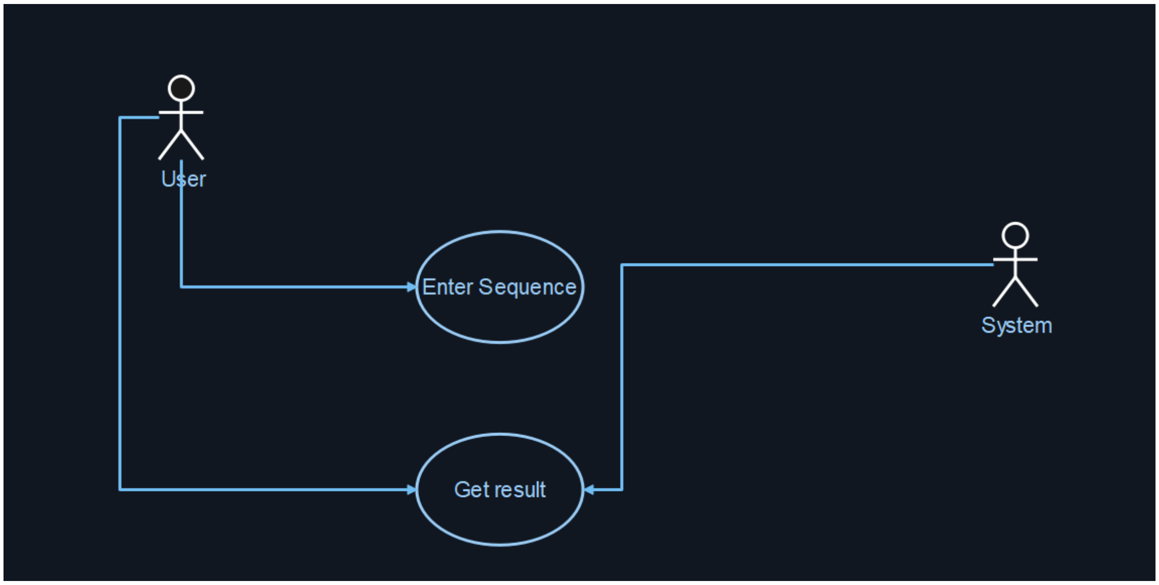


Figure 17

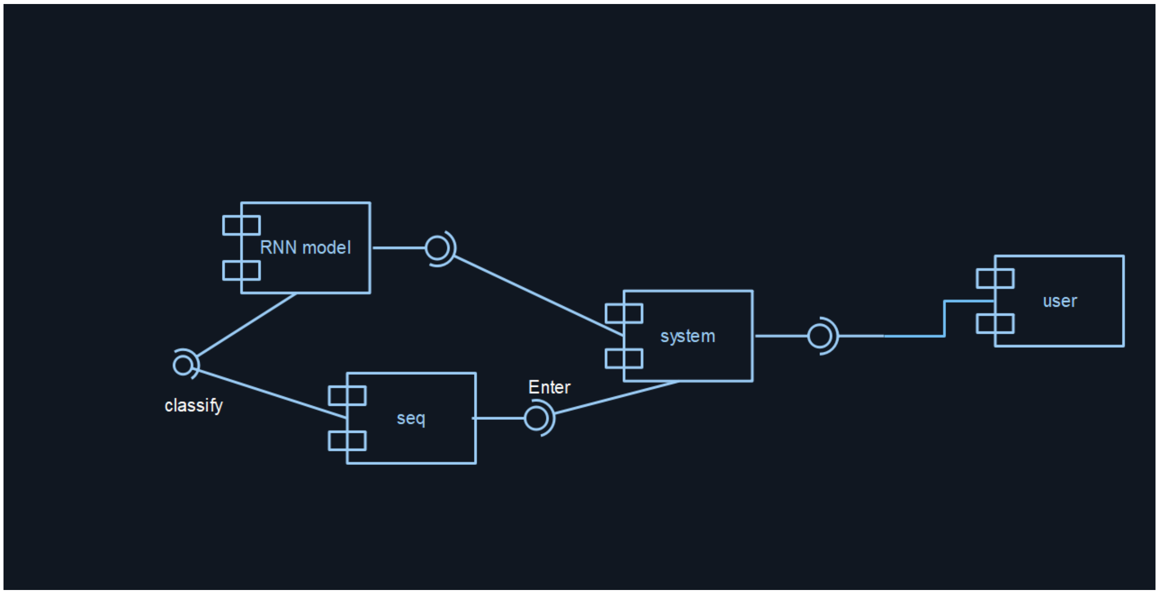
Use case diagram

Chapter Four

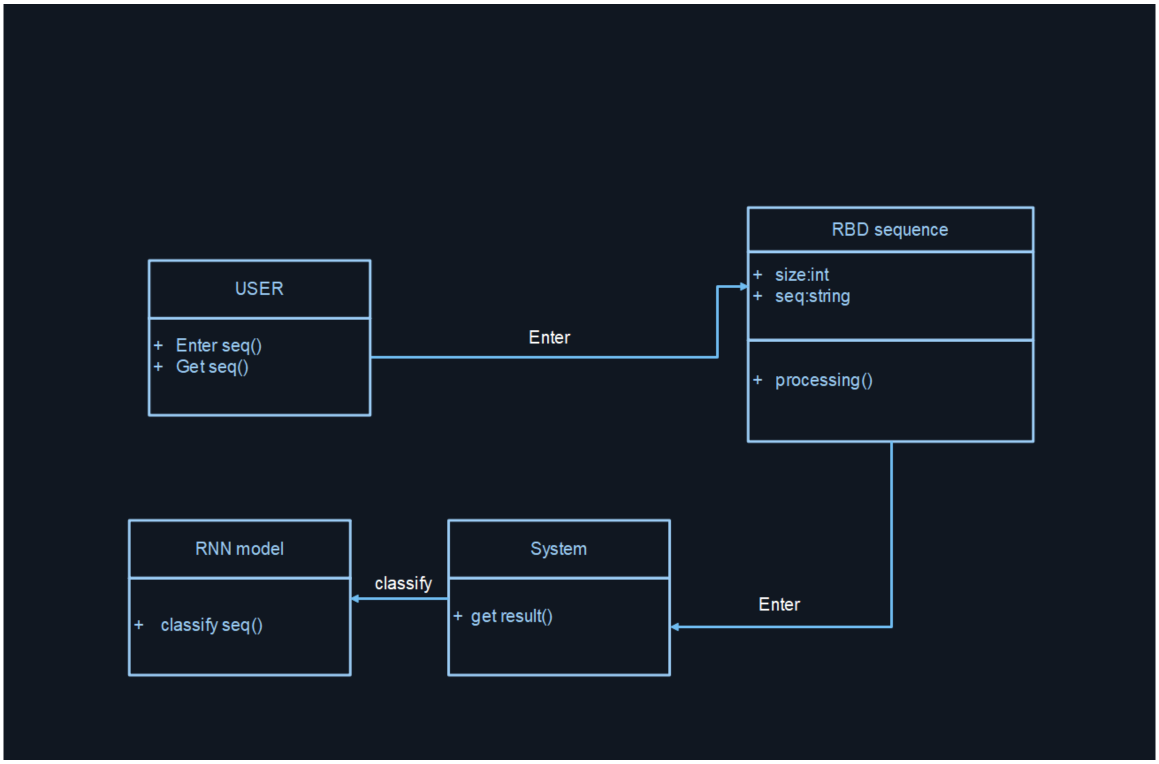
System design

- System component diagram
- System class diagrams
- Sequence diagrams
- System GUI design

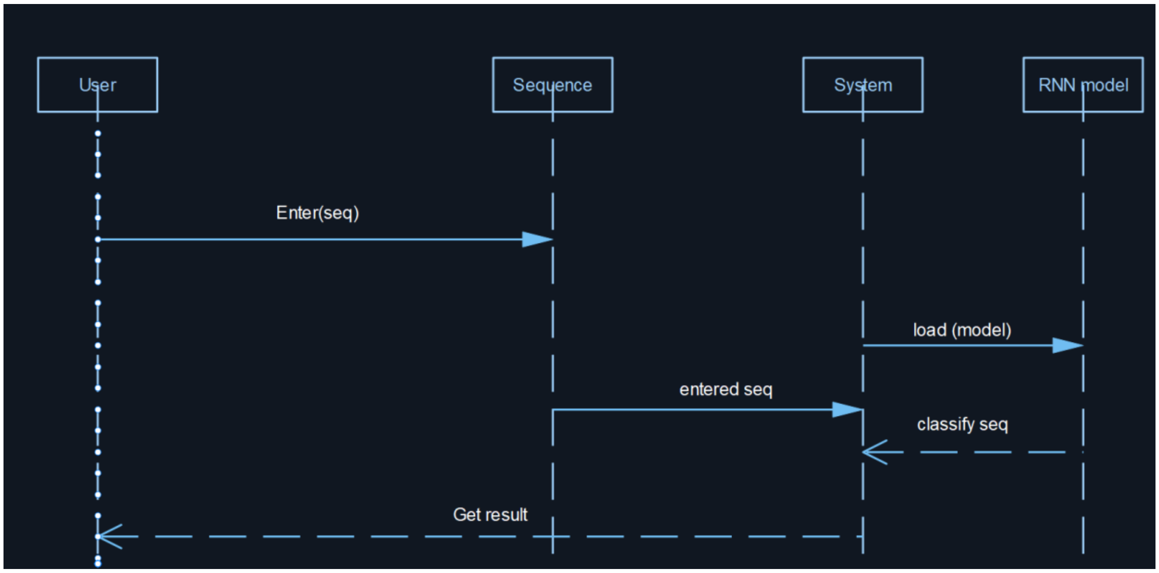
System component diagram:



System class diagram:



Sequence diagram:



Project representation and designing:

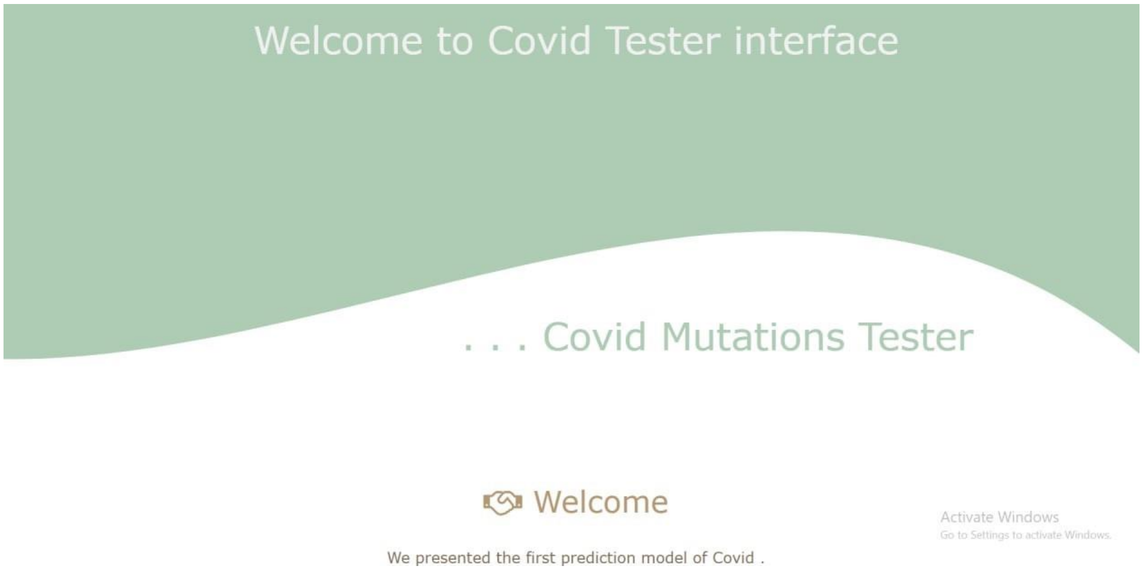
Once model learned from collected and mutated data and tested with mutated data and get high accuracy so model is ready to use.

So, we need to make interface to our project and link it with model so when input entered through interface so it passed to model to predict it infect human cell or not infect.

So, we made machine learning model with high accuracy so if researchers or scientist want to check the RBD infect human cell or not so they don't need to use Swiss model tool to get PDB file then enter it with PDB file of ACE2 (human receptor) to docking tool (HDOCK SERVE) to see if it bind or not Only they need to enter RBD in form of DNA to our website.

So, our website instead of docking tool so our website is interface for Covid detecting tool so its not a system it's a tool instead of docking in covid

System GUI Design:

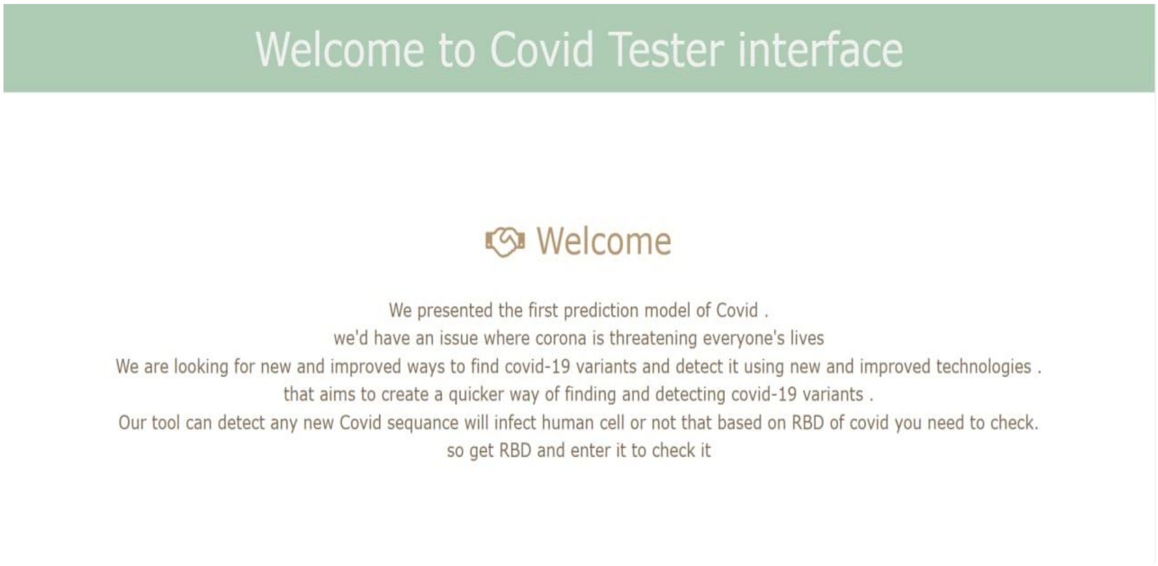


Welcome to Covid Tester interface

. . . Covid Mutations Tester

Welcome

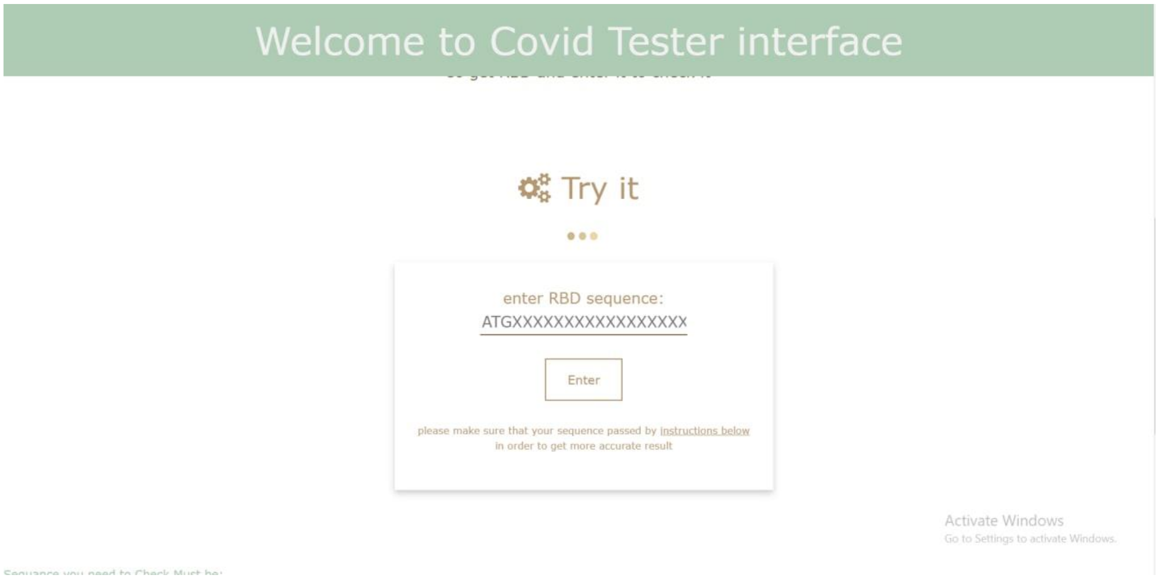
We presented the first prediction model of Covid .



Welcome to Covid Tester interface

Welcome

We presented the first prediction model of Covid .
we'd have an issue where corona is threatening everyone's lives
We are looking for new and improved ways to find covid-19 variants and detect it using new and improved technologies .
that aims to create a quicker way of finding and detecting covid-19 variants .
Our tool can detect any new Covid sequance will infect human cell or not that based on RBD of covid you need to check.
so get RBD and enter it to check it



Welcome to Covid Tester interface

Try it

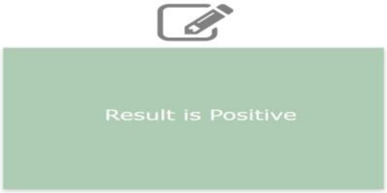
enter RBD sequence:

ATGXXXXXXXXXXXXXXXXX

please make sure that your sequence passed by instructions below in order to get more accurate result

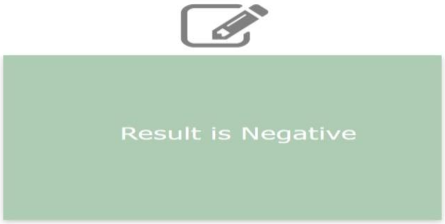
Activate Windows
Go to Settings to activate Windows.

If the mutated RBD binds to human receptor the result will be:



Result is Positive

If the mutated RBD does not bind to human receptor the result will be:



Result is Negative

Chapter Five

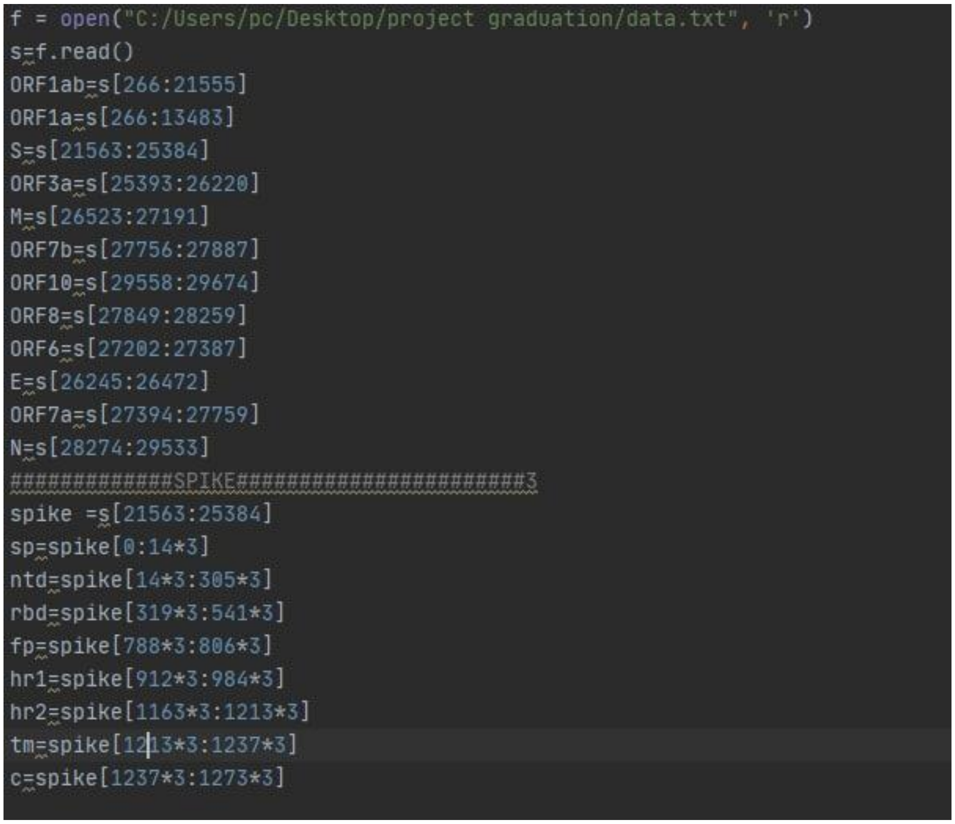
Implementation and testing

Implementation

First:

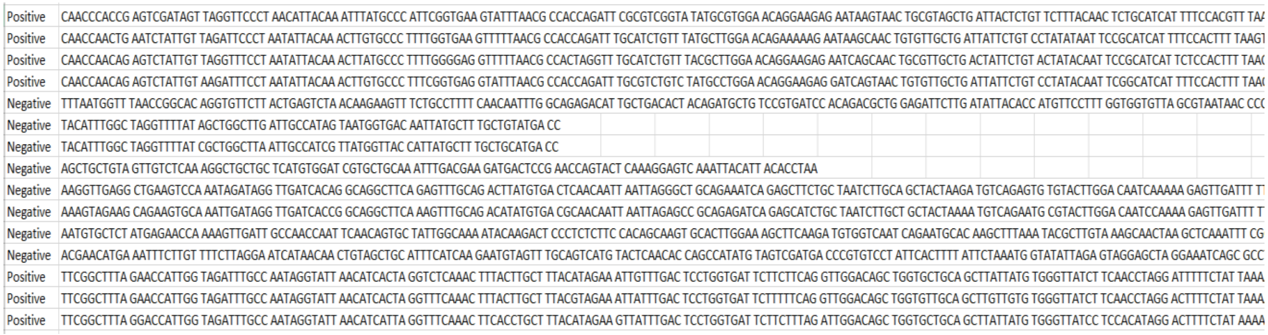
we have implemented Python code that help to take whole genome and known ranges of regions then put it in excel sheet in form of each region DNA and Infront of it positive or negative

(All regions ae negative and RBD is only positive)



```
f = open("C:/Users/pc/Desktop/project graduation/data.txt", 'r')
s=f.read()
ORF1ab=s[266:21555]
ORF1a=s[266:13483]
S=s[21563:25384]
ORF3a=s[25393:26220]
M=s[26523:27191]
ORF7b=s[27756:27887]
ORF10=s[29558:29674]
ORF8=s[27849:28259]
ORF6=s[27202:27387]
E=s[26245:26472]
ORF7a=s[27394:27759]
N=s[28274:29533]
############SPIKE######################3
spike =s[21563:25384]
sp=spike[0:14*3]
ntd=spike[14*3:305*3]
rbd=spike[319*3:541*3]
fp=spike[788*3:806*3]
hr1=spike[912*3:984*3]
hr2=spike[1163*3:1213*3]
tm=spike[1213*3:1237*3]
c=spike[1237*3:1273*3]
```

Output is the following file:



Positive	CAACCCACCG AGTCGATAGT TAGGTTCCCT AACATTACAA ATTTATGCCC ATTCGGTGAA GTATTTAACG CCACCAGATT CGCGTCGGTA TATGCGTGGA ACAGGAAGAG AATAAGTAAC TGCGTAGCTG ATTACTCTGT TCTTTACAAC TCTGCATCAT TTTCCACGTT TAA
Positive	CAACCAACTG AATCTATTGT TAGATTCCCT AATATTACAA ACTTGTGCCC TTTTGGTGAA GTTTTTAACG CCACCAGATT TGCATCTGTT TATGCTTGGA ACAGAAAAAG AATAAGCAAC TGTGTTGCTG ATTATTCTGT CCTATATAAT TCCGCATCAT TTTCCACTTT TAAGT
Positive	CAACCAACAG AGTCTATTGT TAGGTTTCCT AATATTACAA ACTTATGCCC TTTTGGGGAG GTTTTTAACG CCACTAGGTT TGCATCTGTT TACGCTTGGA ACAGGAAGAG AATCAGCAAC TGCGTTGCTG ACTATTCTGT ACTATACAAT TCCGCATCAT TCTCCACTTT TAA
Positive	CAACCAACAG AGTCTATTGT AAGATTTCCT AATATTACAA ACTTGTGCCC TTTCGGTGAG GTATTTAACG CCACCAGATT TGCGTCTGTC TATGCCTGGA ACAGGAAGAG GATCAGTAAC TGTGTTGCTG ATTATTCTGT CCTATACAAT TCGGCATCAT TTTCCACTTT TAA
Negative	TTTAATGGTT TAACCGGCAC AGGTGTTCTT ACTGAGTCTA ACAAGAAGTT TCTGCCTTTT CAACAATTTG GCAGAGACAT TGCTGACACT ACAGATGCTG TCCGTGATCC ACAGACGCTG GAGATTCTTG ATATTACACC ATGTTCCTTT GGTGGTGTTA GCGTAATAAC CC
Negative	TACATTTGGC TAGGTTTTAT AGCTGGCTTG ATTGCCATAG TAATGGTGAC AATTATGCTT TGCTGTATGA CC
Negative	TACATTTGGC TAGGTTTTAT CGCTGGCTTA ATTGCCATCG TTATGGTTAC CATTATGCTT TGCTGCATGA CC
Negative	AGCTGCTGTA GTTGTCTCAA AGGCTGCTGC TCATGTGGAT CGTGCTGCAA ATTTGACGAA GATGACTCCG AACCAGTACT CAAAGGAGTC AAATTACATT ACACCTAA
Negative	AAGGTTGAGG CTGAAGTCCA AATAGATAGG TTGATCACAG GCAGGCTTCA GAGTTTGCAG ACTTATGTGA CTCAACAATT AATTAGGGCT GCAGAAATCA GAGCTTCTGC TAATCTTGCA GCTACTAAGA TGTCAGAGTG TGTACTTGGA CAATCAAAAA GAGTTGATTT T
Negative	AAAGTAGAAG CAGAAGTGCA AATTGATAGG TTGATCACCG GCAGGCTTCA AAGTTTGCAG ACATATGTGA CGCAACAATT AATTAGAGCC GCAGAGATCA GAGCATCTGC TAATCTTGCT GCTACTAAAA TGTCAGAATG CGTACTTGGA CAATCCAAAA GAGTTGATTT T
Negative	AATGTGCTCT ATGAGAACCA AAAGTTGATT GCCAACCAAT TCAACAGTGC TATTGGCAAA ATACAAGACT CCCTCTCTTC CACAGCAAGT GCACTTGGAA AGCTTCAAGA TGTGGTCAAT CAGAATGCAC AAGCTTTAAA TACGCTTGTA AAGCAACTAA GCTCAAATTT CG
Negative	ACGAACATGA AATTTCTTGT TTTCTTAGGA ATCATAACAA CTGTAGCTGC ATTTCATCAA GAATGTAGTT TGCAGTCATG TACTCAACAC CAGCCATATG TAGTCGATGA CCCGTGTCCT ATTCACTTTT ATTCTAAATG GTATATTAGA GTAGGAGCTA GGAAATCAGC GCC
Positive	TTCGGCTTTA GAACCATTGG TAGATTTGCC AATAGGTATT AACATCACTA GGTCTCAAAC TTTACTTGCT TTACATAGAA ATTGTTTGAC TCCTGGTGAT TCTTCTTCAG GTTGGACAGC TGGTGCTGCA GCTTATTATG TGGGTTATCT TCAACCTAGG ATTTTTCTAT TAAA
Positive	TTCGGCTTTA GAACCATTGG TAGATTTGCC AATAGGTATT AACATCACTA GGTTTCAAAC TTTACTTGCT TTACGTAGAA ATTATTTGAC TCCTGGTGAT TCTTTTTCAG GTTGGACAGC TGGTGTTGCA GCTTGTTGTG TGGGTTATCT TCAACCTAGG ACTTTTCTAT TAAA
Positive	TTCGGCTTTA GGACCATTGG TAGATTTGCC AATAGGTATT AACATCATTA GGTTTCAAAC TTCACCTGCT TTACATAGAA GTTATTTGAC TCCTGGTGAT TCTTCTTTAG ATTGGACAGC TGGTGCTGCA GCTTATTATG TGGGTTATCC TCCACATAGG ACTTTTCTAT AAAA

Model Architecture

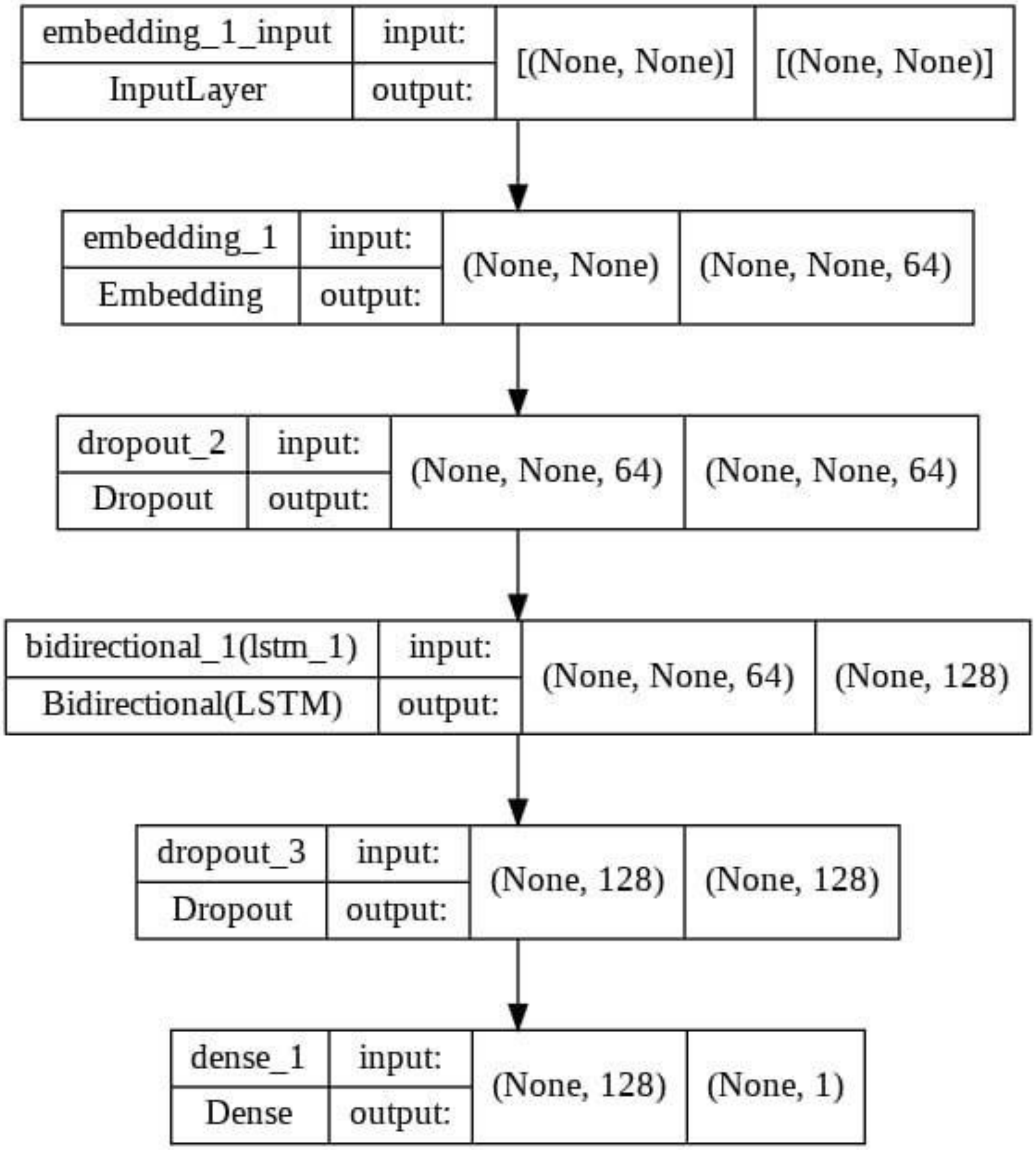
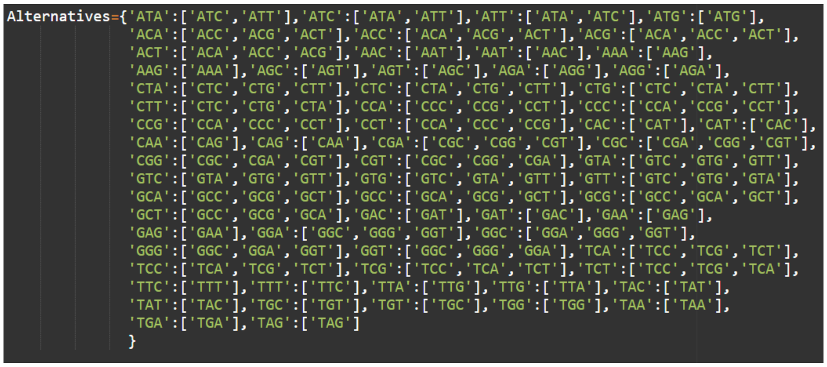


Figure 18

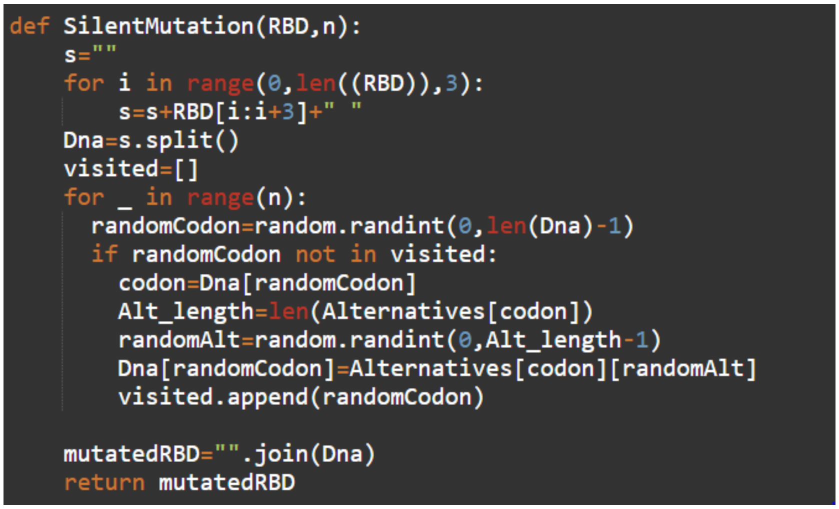
This figure shows model architecture

Second:

Silent Mutation:



```
Alternatives={'ATA':['ATC','ATT'],'ATC':['ATA','ATT'],'ATT':['ATA','ATC'],'ATG':['ATG'],
              'ACA':['ACC','ACG','ACT'],'ACC':['ACA','ACG','ACT'],'ACG':['ACA','ACC','ACT'],
              'ACT':['ACA','ACC','ACG'],'AAC':['AAT'],'AAT':['AAC'],'AAA':['AAG'],
              'AAG':['AAA'],'AGC':['AGT'],'AGT':['AGC'],'AGA':['AGG'],'AGG':['AGA'],
              'CTA':['CTC','CTG','CTT'],'CTC':['CTA','CTG','CTT'],'CTG':['CTC','CTA','CTT'],
              'CTT':['CTC','CTG','CTA'],'CCA':['CCC','CCG','CCT'],'CCC':['CCA','CCG','CCT'],
              'CCG':['CCA','CCC','CCT'],'CCT':['CCA','CCC','CCG'],'CAC':['CAT'],'CAT':['CAC'],
              'CAA':['CAG'],'CAG':['CAA'],'CGA':['CGC','CGG','CGT'],'CGC':['CGA','CGG','CGT'],
              'CGG':['CGC','CGA','CGT'],'CGT':['CGC','CGG','CGA'],'GTA':['GTC','GTG','GTT'],
              'GTC':['GTA','GTG','GTT'],'GTG':['GTC','GTA','GTT'],'GTT':['GTC','GTG','GTA'],
              'GCA':['GCC','GCG','GCT'],'GCC':['GCA','GCG','GCT'],'GCG':['GCC','GCA','GCT'],
              'GCT':['GCC','GCG','GCA'],'GAC':['GAT'],'GAT':['GAC'],'GAA':['GAG'],
              'GAG':['GAA'],'GGA':['GGC','GGG','GGT'],'GGC':['GGA','GGG','GGT'],
              'GGG':['GGC','GGA','GGT'],'GGT':['GGC','GGG','GGA'],'TCA':['TCC','TCG','TCT'],
              'TCC':['TCA','TCG','TCT'],'TCG':['TCC','TCA','TCT'],'TCT':['TCC','TCG','TCA'],
              'TTC':['TTT'],'TTT':['TTC'],'TTA':['TTG'],'TTG':['TTA'],'TAC':['TAT'],
              'TAT':['TAC'],'TGC':['TGT'],'TGT':['TGC'],'TGG':['TGG'],'TAA':['TAA'],
              'TGA':['TGA'],'TAG':['TAG']
              }
```



```
def SilentMutation(RBD,n):
    s=""
    for i in range(0,len((RBD)),3):
        s=s+RBD[i:i+3]+" "
    Dna=s.split()
    visited=[]
    for _ in range(n):
      randomCodon=random.randint(0,len(Dna)-1)
      if randomCodon not in visited:
        codon=Dna[randomCodon]
        Alt_length=len(Alternatives[codon])
        randomAlt=random.randint(0,Alt_length-1)
        Dna[randomCodon]=Alternatives[codon][randomAlt]
        visited.append(randomCodon)

    mutatedRBD="".join(Dna)
    return mutatedRBD
```

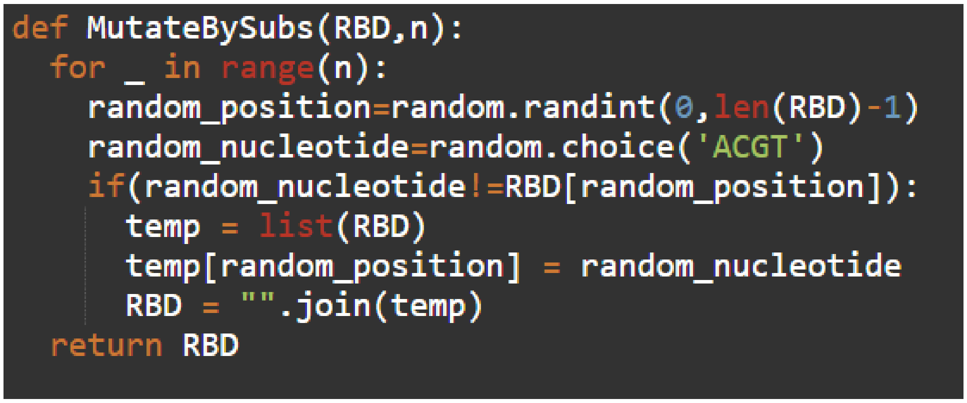
First we made a dictionary called "Alternatives" which holds every codon and a list of its alternatives that will be translated to the same amino acid , for ex: GGA has alternatives (['GGC','GGG','GGT']) as they all are translated to Gly.

Then we split the RBD into list of codons.

We created a list visited to hold the positions at which the nucleotide is changed.

Then we get random codon from the list of codons of the RBD and check if it is present in the list visited, if not, we get the number of the codon's alternatives and we choose a random alternative then we replace the codon with this alternative.

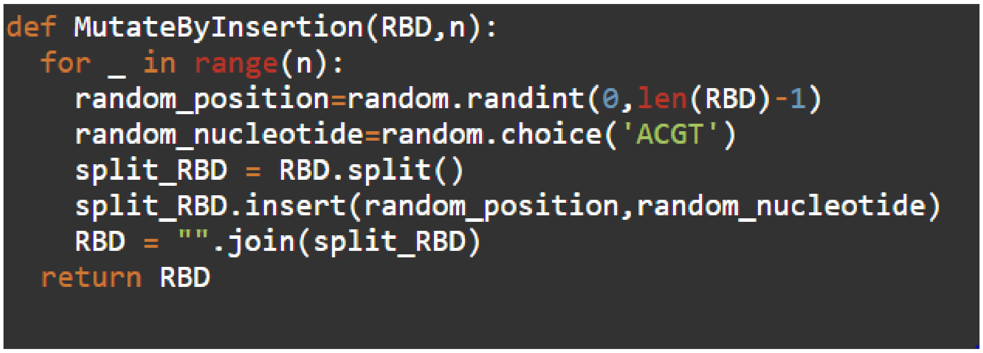
Substitution Mutation:



```
def MutateBySubs(RBD,n):
  for _ in range(n):
    random_position=random.randint(0,len(RBD)-1)
    random_nucleotide=random.choice('ACGT')
    if(random_nucleotide!=RBD[random_position]):
      temp = list(RBD)
      temp[random_position] = random_nucleotide
      RBD = "".join(temp)
  return RBD
```

In case of substitution, we get random nucleotide from (A C G T) then we check that the nucleotide of the random position we got is not the same as the random nucleotide, if yes, we substitute the original nucleotide by the random nucleotide.

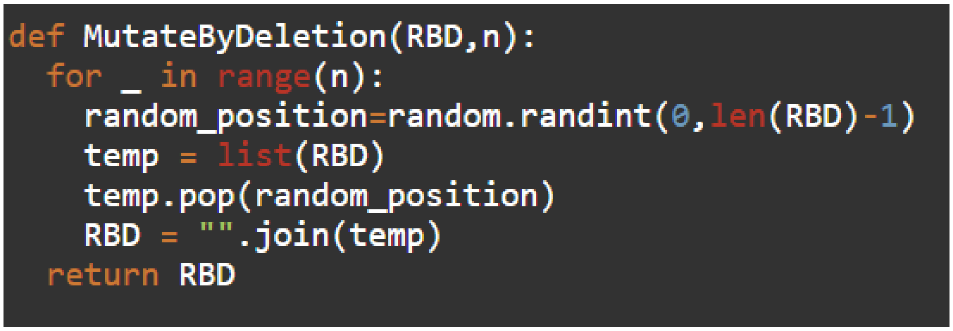
Insertion Mutation:



```
def MutateByInsertion(RBD,n):
  for _ in range(n):
    random_position=random.randint(0,len(RBD)-1)
    random_nucleotide=random.choice('ACGT')
    split_RBD = RBD.split()
    split_RBD.insert(random_position,random_nucleotide)
    RBD = "".join(split_RBD)
  return RBD
```

In case of insertion, we get random nucleotide from (A C G T) then insert it in the random position we got.

Deletion Mutation:



```
def MutateByDeletion(RBD,n):
  for _ in range(n):
    random_position=random.randint(0,len(RBD)-1)
    temp = list(RBD)
    temp.pop(random_position)
    RBD = "".join(temp)
  return RBD
```

In case of deletion, we delete the nucleotide of the random position we got.

Third:

At first, we take four variants as train data in form of excel sheet then add to it all mutated discovered until now on NCBI the add mutated data of four variant that we created.

So, after all of that we need to make sure that all of these data in excel sheet have no duplicates to avoid overfitting of model

So, we need to save DNA in dictionary as key cause key of dictionary is unique of each region

```
import csv

#Dictionry needed to save region key on it

dit= {}

count=0

rbd='AGATTTCCTA ATATTACAAA CTTGTGCCCT TTTGGTGAAG
TTTTTAACGC CACCAGATTT GCATCTGTTT ATGCTTGGAA CAGGAAGAGA
ATCAGCAACT GTGTTGCTGA TTATTCTGTC CTATATAATT CCGCATCATT
TTCCACTTTT AAGTGTTATG GAGTGTCTCC TACTAAATTA AATGATCTCT
GCTTTACTAA TGTCTATGCA GATTCATTTG TAATTAGAGG TGATGAAGTC
AGACAAATCG CTCCAGGGCA AACTGGAAAG ATTGCTGATT ATAATTATAA
ATTACCAGAT '

x=' '

with open("/content/sample_data/test3_3_.csv", 'r') as csvfile:

    reader = csv.reader(csvfile, delimiter=',')

    #read file row by row

    for row in reader:

        # labels will have word of + or -

        if (row[0]=='' or row[0]=='S(region)'):

          count+=1

          dit[row[0]]=count

         #add splited rbd in case of rbd region

          x+=row[0][len(row[0])-2]
```

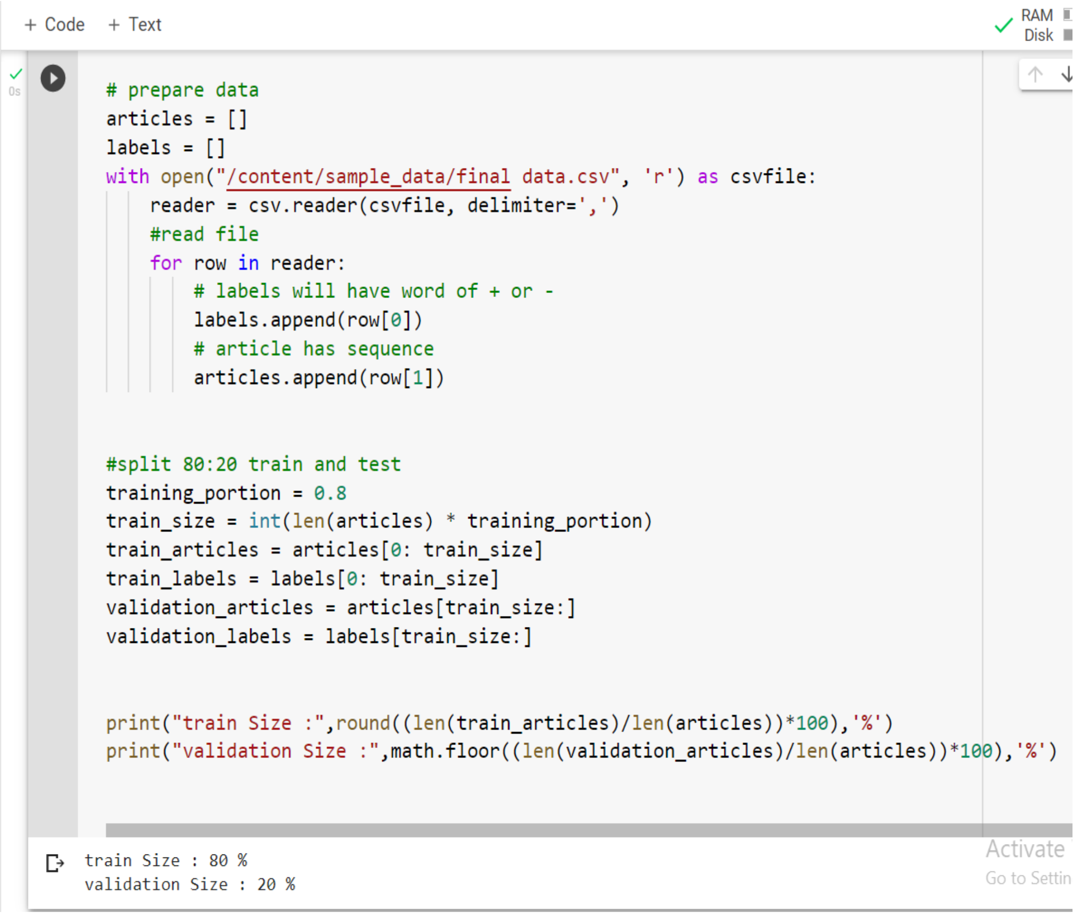
```
#Split RBD again after remove duplicates
#Split into 67 segments each broken into words of length 10
RBD={}
for i in dit.keys():
  seqweb=i
  seqget = [seqweb[i:i+67] for i in range(0, len(seqweb), 67)]
  c=0
  for i in seqget:
    c+=1
    x=[i[i2:i2+10] for i2 in range(0, len(i), 10)]
    l=''
    for k in x:
      l+=k+" "
    RBD[l]=c
```

Fourth:

Finally, we collect our training data so we need to build our model (RNN model), that’s imports (libraries that help us in coding python)

```
#our imports
import csv
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, LSTM, Dropout, Activation,
Embedding, Bidirectional,SimpleRNN
import nltk
from nltk.corpus import stopwords
import math
```

Then we need to read file and save data in list and split it into train and validation data 80:20



```
# prepare data
articles = []
labels = []
with open("/content/sample_data/final data.csv", 'r') as csvfile:
    reader = csv.reader(csvfile, delimiter=',')
    #read file
    for row in reader:
        # labels will have word of + or -
        labels.append(row[0])
        # article has sequence
        articles.append(row[1])


#split 80:20 train and test
training_portion = 0.8
train_size = int(len(articles) * training_portion)
train_articles = articles[0: train_size]
train_labels = labels[0: train_size]
validation_articles = articles[train_size:]
validation_labels = labels[train_size:]


print("train Size :",round((len(train_articles)/len(articles))*100),'%')
print("validation Size :",math.floor((len(validation_articles)/len(articles))*100),'%')
```

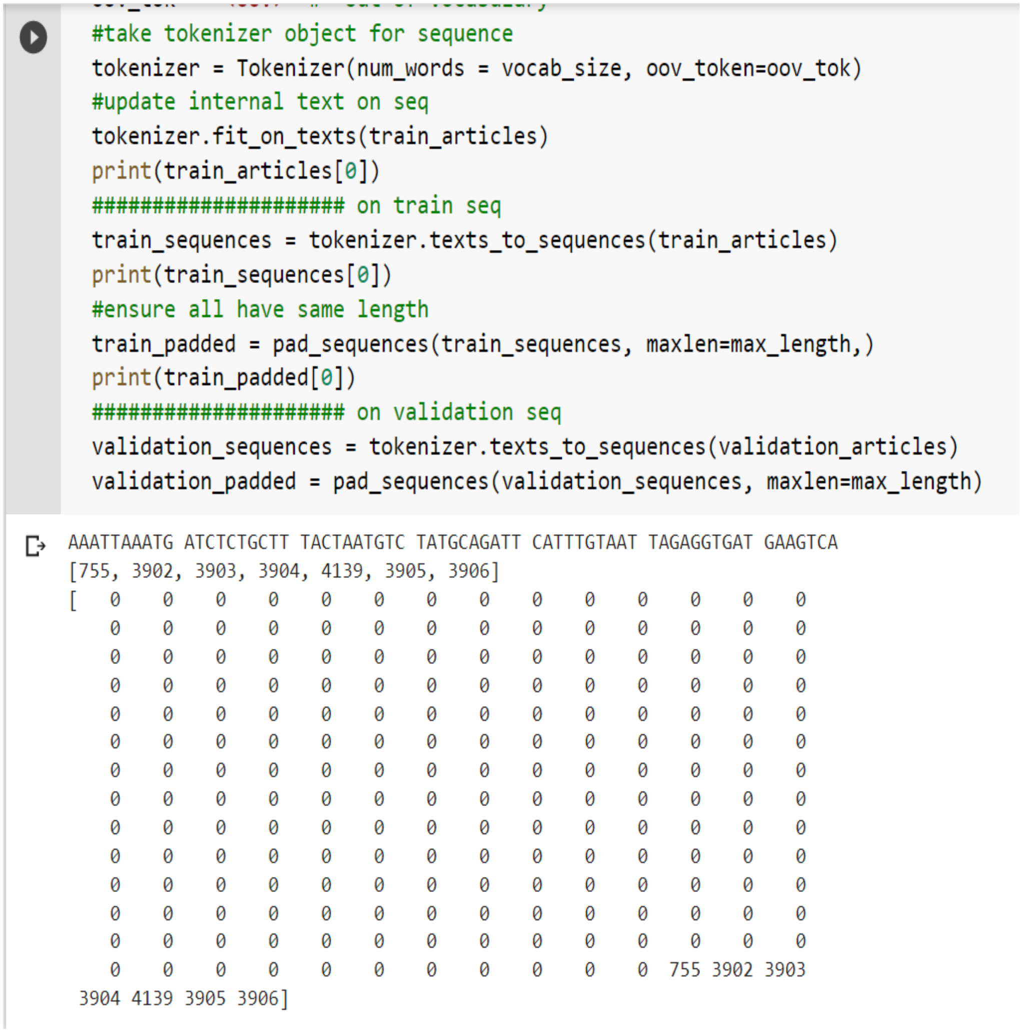
```
train Size : 80 %
validation Size : 20 %
```

Then we need to use tokenizer cause we working on text and need to be able to put in way help model to learn it.

So, need specify first vocab size and max length of it

```
vocab_size = 6200

max_length = 200

oov_tok = '<OOV>' #  Out of Vocabulary
```

Then we need to use tokenizer to convert our words into numerical values that can be processed by a machine learning model and make padding to it to ensure all have same length so result is list of large lists with same length



```
#take tokenizer object for sequence
tokenizer = Tokenizer(num_words = vocab_size, oov_token=oov_tok)
#update internal text on seq
tokenizer.fit_on_texts(train_articles)
print(train_articles[0])
##################### on train seq
train_sequences = tokenizer.texts_to_sequences(train_articles)
print(train_sequences[0])
#ensure all have same length
train_padded = pad_sequences(train_sequences, maxlen=max_length,)
print(train_padded[0])
##################### on validation seq
validation_sequences = tokenizer.texts_to_sequences(validation_articles)
validation_padded = pad_sequences(validation_sequences, maxlen=max_length)
```

```
AAATTAAATG ATCTCTGCTT TACTAATGTC TATGCAGATT CATTTGTAAT TAGAGGTGAT GAAGTCA
[755, 3902, 3903, 3904, 4139, 3905, 3906]
[   0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0  755 3902 3903
 3904 4139 3905 3906]
```

We also need to make this for both validation and training data to be both numeric value that’s can be trained by our RNN model.

Now we adjust training and validation sequence we need to adjust label also. In order to adjust label also we need to write label in form of zeros and ones not positive and negative to be binary cross entropy model which produce only one probability if it is greater than 0.5 so it is positive else its negative

So we use data frame to use get dummies that make that easier by put zeros and ones instead of two words

So, help to convert categorical data into dummy.

And we need to make that to training label and also validation label

Cause we need both Must be represented in 0 instead of negative and 1 instead of positive

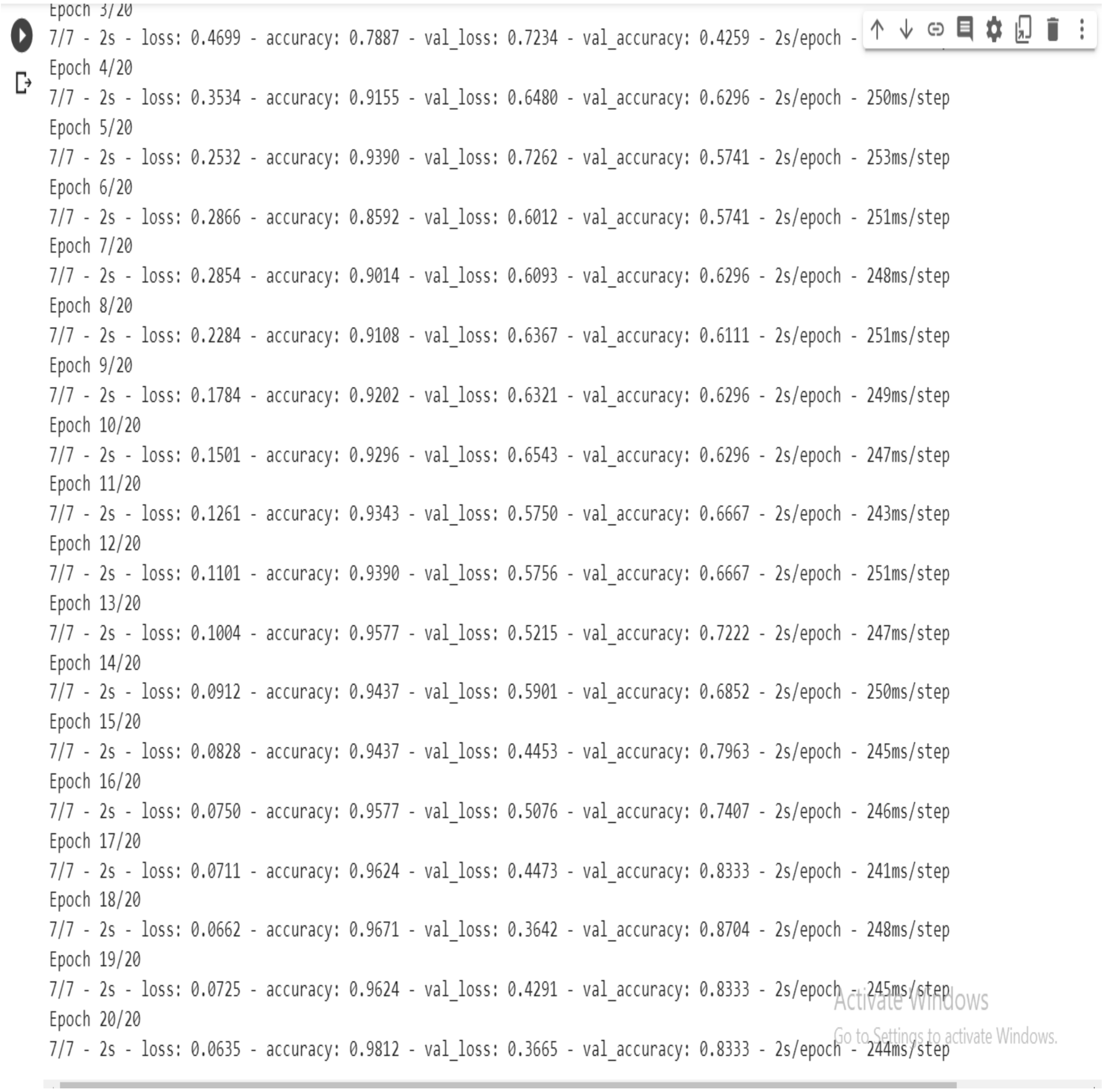
```
#labels  inform of 0(-) and 1(+)

data_frame = pd.DataFrame(train_labels, columns=["type"])

# convert categorical data into dummy

df_one = pd.get_dummies(data_frame["type"])

# bind two columns with type

df_two = pd.concat((df_one, data_frame), axis=1)

# drop column type and negative(give negative one)

df_two = df_two.drop(["type"], axis=1)

df_two = df_two.drop(["Negative"], axis=1)

#rename positive column

trainlabelresult = df_two.rename(columns={"Positive": "type"})

############################# make for validation also

data_frame2 = pd.DataFrame(validation_labels, columns=["type"])

df_one2 = pd.get_dummies(data_frame2["type"])

df_two2 = pd.concat((df_one2, data_frame2), axis=1)

df_two2 = df_two2.drop(["type"], axis=1)

df_two2 = df_two2.drop(["Negative"], axis=1)

validationlabelresult = df_two2.rename(columns={"Positive": "type"})
```

Then we build our model using sequential and specify hidden layer and learning rate and number of epoch and activation function and pass all these data to learn model it

```
#our model

embedding_dim = 64

model = Sequential()

model.add(Embedding(vocab_size,embedding_dim))

model.add(Dropout(0.2))

model.add(Bidirectional(LSTM(embedding_dim)))

model.add(Dropout(0.2))

model.add(Dense(1,activation='sigmoid'))

opt = tf.keras.optimizers.Adam(lr=0.002, decay=1e-6)

model.compile(loss='binary_crossentropy',optimizer=opt,metrics=['accuracy'])

num_epochs = 20

history = model.fit(train_padded, trainlabelresult, epochs=num_epochs,validation_data=(validation_padded, validationlabelresult),verbose=2)
```

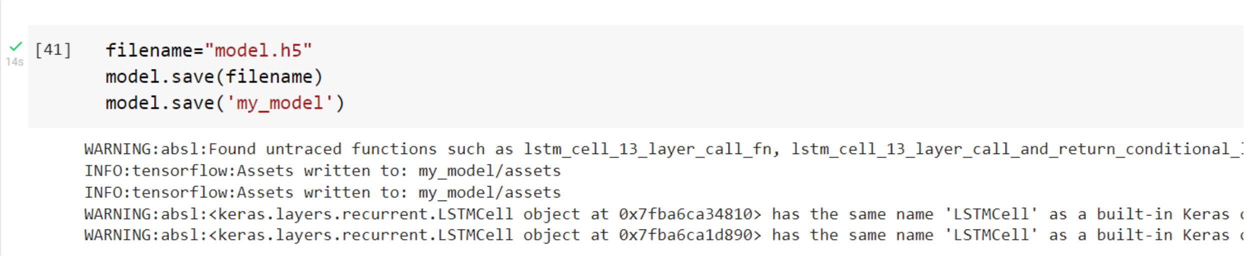
Then we run this model to let it learn from data passed to it and

Accuracy will reach 90 % and that is high and good result so that’s mean the model learn in good manner and that will be represented during running of epoch and will attached below



```
Epoch 3/20
7/7 - 2s - loss: 0.4699 - accuracy: 0.7887 - val_loss: 0.7234 - val_accuracy: 0.4259 - 2s/epoch -
Epoch 4/20
7/7 - 2s - loss: 0.3534 - accuracy: 0.9155 - val_loss: 0.6480 - val_accuracy: 0.6296 - 2s/epoch - 250ms/step
Epoch 5/20
7/7 - 2s - loss: 0.2532 - accuracy: 0.9390 - val_loss: 0.7262 - val_accuracy: 0.5741 - 2s/epoch - 253ms/step
Epoch 6/20
7/7 - 2s - loss: 0.2866 - accuracy: 0.8592 - val_loss: 0.6012 - val_accuracy: 0.5741 - 2s/epoch - 251ms/step
Epoch 7/20
7/7 - 2s - loss: 0.2854 - accuracy: 0.9014 - val_loss: 0.6093 - val_accuracy: 0.6296 - 2s/epoch - 248ms/step
Epoch 8/20
7/7 - 2s - loss: 0.2284 - accuracy: 0.9108 - val_loss: 0.6367 - val_accuracy: 0.6111 - 2s/epoch - 251ms/step
Epoch 9/20
7/7 - 2s - loss: 0.1784 - accuracy: 0.9202 - val_loss: 0.6321 - val_accuracy: 0.6296 - 2s/epoch - 249ms/step
Epoch 10/20
7/7 - 2s - loss: 0.1501 - accuracy: 0.9296 - val_loss: 0.6543 - val_accuracy: 0.6296 - 2s/epoch - 247ms/step
Epoch 11/20
7/7 - 2s - loss: 0.1261 - accuracy: 0.9343 - val_loss: 0.5750 - val_accuracy: 0.6667 - 2s/epoch - 243ms/step
Epoch 12/20
7/7 - 2s - loss: 0.1101 - accuracy: 0.9390 - val_loss: 0.5756 - val_accuracy: 0.6667 - 2s/epoch - 251ms/step
Epoch 13/20
7/7 - 2s - loss: 0.1004 - accuracy: 0.9577 - val_loss: 0.5215 - val_accuracy: 0.7222 - 2s/epoch - 247ms/step
Epoch 14/20
7/7 - 2s - loss: 0.0912 - accuracy: 0.9437 - val_loss: 0.5901 - val_accuracy: 0.6852 - 2s/epoch - 250ms/step
Epoch 15/20
7/7 - 2s - loss: 0.0828 - accuracy: 0.9437 - val_loss: 0.4453 - val_accuracy: 0.7963 - 2s/epoch - 245ms/step
Epoch 16/20
7/7 - 2s - loss: 0.0750 - accuracy: 0.9577 - val_loss: 0.5076 - val_accuracy: 0.7407 - 2s/epoch - 246ms/step
Epoch 17/20
7/7 - 2s - loss: 0.0711 - accuracy: 0.9624 - val_loss: 0.4473 - val_accuracy: 0.8333 - 2s/epoch - 241ms/step
Epoch 18/20
7/7 - 2s - loss: 0.0662 - accuracy: 0.9671 - val_loss: 0.3642 - val_accuracy: 0.8704 - 2s/epoch - 248ms/step
Epoch 19/20
7/7 - 2s - loss: 0.0725 - accuracy: 0.9624 - val_loss: 0.4291 - val_accuracy: 0.8333 - 2s/epoch - 245ms/step
Epoch 20/20
7/7 - 2s - loss: 0.0635 - accuracy: 0.9812 - val_loss: 0.3665 - val_accuracy: 0.8333 - 2s/epoch - 244ms/step
```

And now we need to save model to be able to connect it with the interface so we will save it by



```
filename="model.h5"
model.save(filename)
model.save('my_model')
```

```
WARNING:absl:Found untraced functions such as lstm_cell_13_layer_call_fn, lstm_cell_13_layer_call_and_return_conditional_
INFO:tensorflow:Assets written to: my_model/assets
INFO:tensorflow:Assets written to: my_model/assets
WARNING:absl:<keras.layers.recurrent.LSTMCell object at 0x7fba6ca34810> has the same name 'LSTMCell' as a built-in Keras
WARNING:absl:<keras.layers.recurrent.LSTMCell object at 0x7fba6ca1d890> has the same name 'LSTMCell' as a built-in Keras
```

And now we need to connect this saved model with our website.

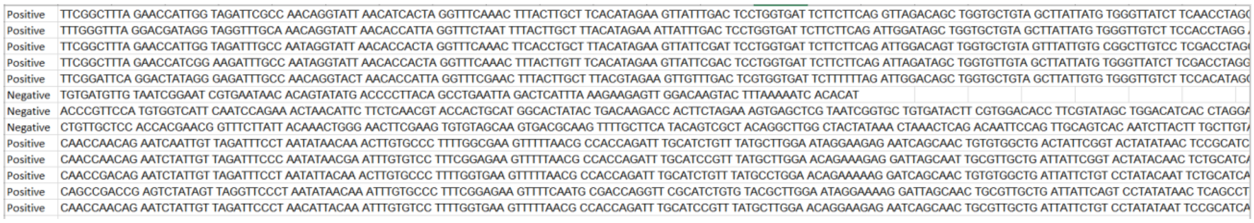
So, we will use Powerful flask to connect with model (by load it)and render our website interface (by render template)

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def covid():
    #load model
     model=tf.keras.models.load_model('my_model')
    # take input from website to make it predicted
     input=request.form.get("seq")
    pred=model.predict(input)
    # rednder result in website
    return render_template("result.html")
if __name__ == '__main__':
    app.run()
```

Testing

First:

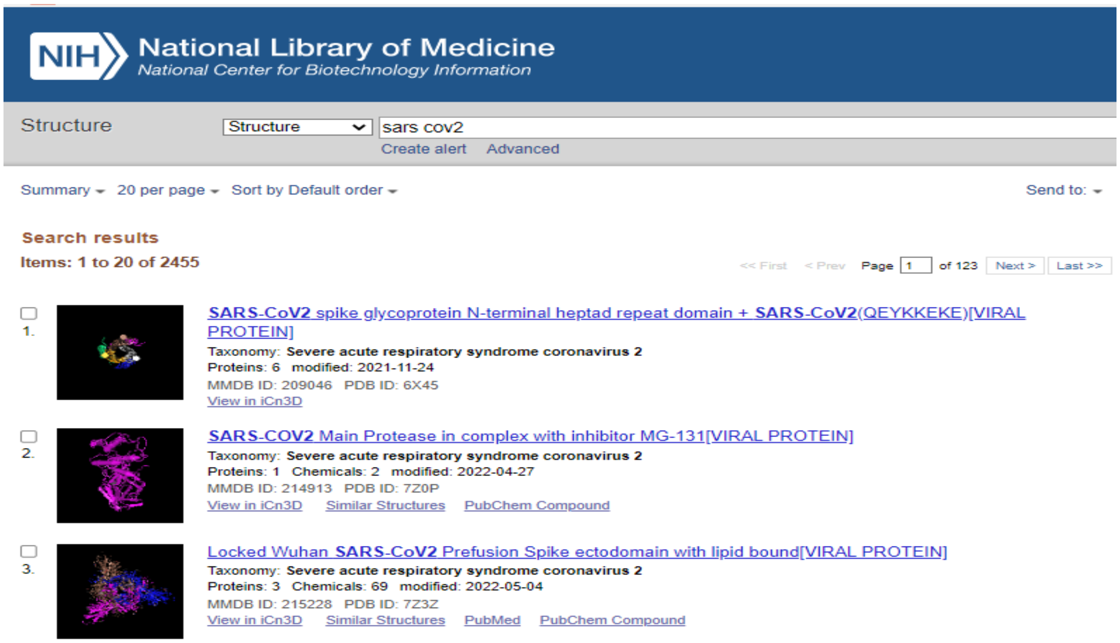
We tested using Mutated RBDs that we got through making silent mutation to each RBD of the 4 variants.



Positive	TTCGGCTTTA GAACCATTGG TAGATTCGCC AACAGGTATT AACATCACTA GGTTTCAAAC TTTACTTGCT TCACATAGAA GTTATTTGAC TCCTGGTGAT TCTTCTTCAG GTTAGACAGC TGGTGCTGTA GCTTATTATG TGGGTTATCT TCAACCTAG
Positive	TTTGGGTTTA GGACGATAGG TAGGTTTGCA AACAGGTATT AACACCATTA GGTTTCTAAT TTTACTTGCT TTACATAGAA ATTATTTGAC TCCTGGTGAT TCTTCTTCAG ATTGGATAGC TGGTGCTGTA GCTTATTATG TGGGTTGTCT TCCACCTAGG
Positive	TTCGGCTTTA GAACCATTGG TAGATTTGCC AATAGGTATT AACACCACTA GGTTTCAAAC TTCACCTGCT TTACATAGAA GTTATTCGAT TCCTGGTGAT TCTTCTTCAG ATTGGACAGT TGGTGCTGTA GTTTATTGTG CGGCTTGTCC TCGACCTAG
Positive	TTCGGCTTTA GAACCATCGG AAGATTTGCC AATAGGTATT AACACCACTA GGTTTCAAAC TTTACTTGTT TCACATAGAA GTTATTCGAC TCCTGGTGAT TCTTCTTCAG ATTAGATAGC TGGTGTTGTA GCTTATTATG TGGGTTATCT TCGACCTAGG
Positive	TTCGGATTCA GGACTATAGG GAGATTTGCC AACAGGTACT AACACCATTA GGTTTCGAAC TTTACTTGCT TTACGTAGAA GTTGTTTGAC TCGTGGTGAT TCTTTTTTAG ATTGGACAGC TGGTGCTGTA GCTTATTGTG TGGGTTGTCT TCCACATAG
Negative	TGTGATGTTG TAATCGGAAT CGTGAATAAC ACAGTATATG ACCCCTTACA GCCTGAATTA GACTCATTTA AAGAAGAGTT GGACAAGTAC TTTAAAAATC ACACAT
Negative	ACCCGTTCCA TGTGGTCATT CAATCCAGAA ACTAACATTC TTCTCAACGT ACCACTGCAT GGCACTATAC TGACAAGACC ACTTCTAGAA AGTGAGCTCG TAATCGGTGC TGTGATACTT CGTGGACACC TTCGTATAGC TGGACATCAC CTAGGA
Negative	CTGTTGCTCC ACCACGAACG GTTTCTTATT ACAAACTGGG AACTTCGAAG TGTGTAGCAA GTGACGCAAG TTTTGCTTCA TACAGTCGCT ACAGGCTTGG CTACTATAAA CTAAACTCAG ACAATTCCAG TTGCAGTCAC AATCTTACTT TGCTTGTA
Positive	CAACCAACAG AATCAATTGT TAGATTTCCT AATATAACAA ACTTGTGCCC TTTTGGCGAA GTTTTTAACG CCACCAGATT TGCATCTGTT TATGCTTGGA ATAGGAAGAG AATCAGCAAC TGTGTGGCTG ACTATTCGGT ACTATATAAC TCCGCATC
Positive	CAACCAACAG AATCTATTGT TAGATTTCCC AATATAACGA ATTTGTGTCC TTTCGGAGAA GTTTTTAACG CCACCAGATT TGCATCCGTT TATGCTTGGA ACAGAAAGAG GATTAGCAAT TGCGTTGCTG ATTATTCGGT ACTATACAAC TCTGCATCA
Positive	CAACCGACAG AATCTATTGT TAGATTTCCT AATATTACAA ACTTGTGCCC TTTTGGTGAA GTTTTTAACG CCACCAGATT TGCATCTGTT TATGCCTGGA ACAGAAAAAG GATCAGCAAC TGTGTGGCTG ATTATTCTGT CCTATACAAT TCTGCATCA
Positive	CAGCCGACCG AGTCTATAGT TAGGTTCCCT AATATAACAA ATTTGTGCCC TTTCGGAGAA GTTTTCAATG CGACCAGGTT CGCATCTGTG TACGCTTGGA ATAGGAAAAG GATTAGCAAC TGCGTTGCTG ATTATTCAGT CCTATATAAC TCAGCCT
Positive	CAACCAACAG AATCTATTGT TAGATTCCCT AACATTACAA ATTTGTGTCC TTTTGGTGAA GTTTTTAACG CCACCAGATT TGCATCCGTT TATGCTTGGA ACAGGAAGAG AATCAGCAAC TGCGTTGCTG ATTATTCTGT CCTATATAAT TCCGCATCA

Second:

We searched for SARS-COV 2 in NCBI and we downloaded files of that have Spike Glycoprotein with other light and heavy chains then we download the PDB file of the spike protein only , then we used pdb2fasta tool to get it's protein then we got it's DNA and entered it as input for RNN model and it predicted it correctly (+ve).



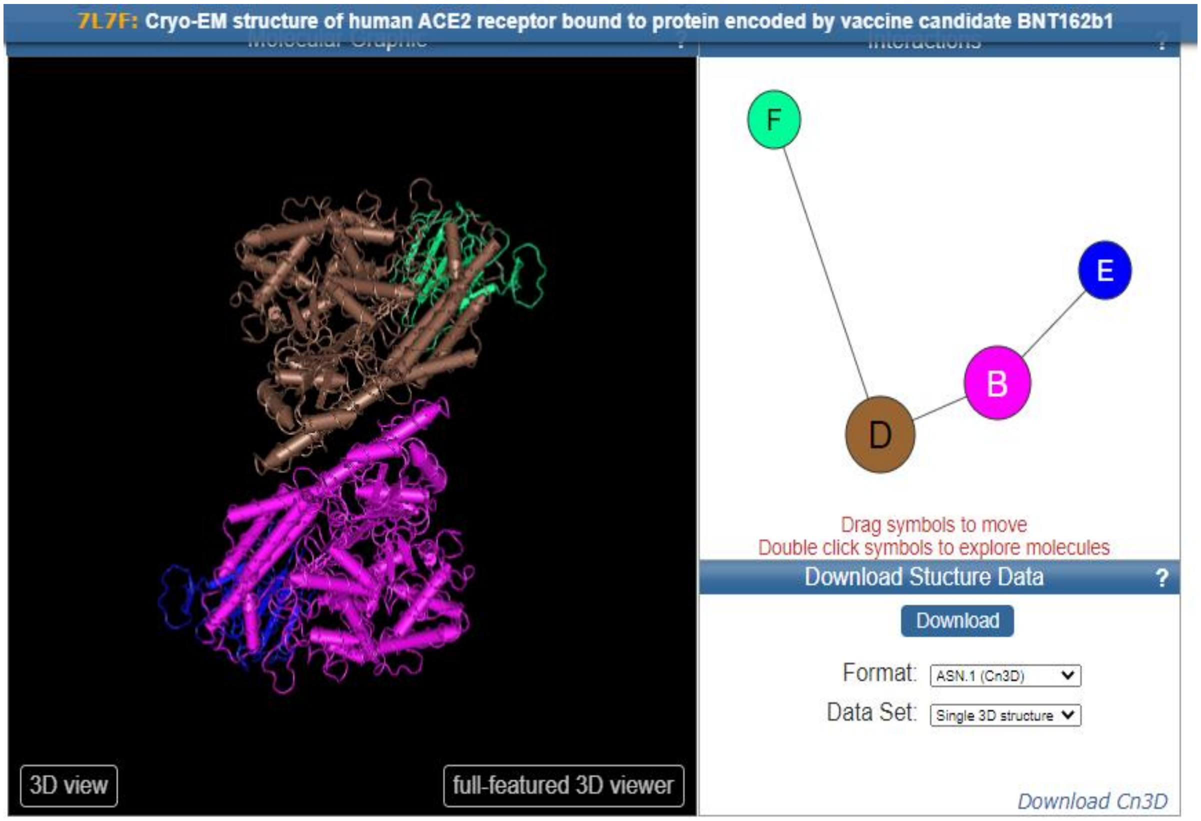
Search results

Items: 1 to 20 of 2455

1. SARS-CoV2 spike glycoprotein N-terminal heptad repeat domain + SARS-CoV2(QEYKKEKE)[VIRAL PROTEIN]
Taxonomy: Severe acute respiratory syndrome coronavirus 2
Proteins: 6 modified: 2021-11-24
MMDB ID: 209046 PDB ID: 6X45
View in iCn3D

2. SARS-COV2 Main Protease in complex with inhibitor MG-131[VIRAL PROTEIN]
Taxonomy: Severe acute respiratory syndrome coronavirus 2
Proteins: 1 Chemicals: 2 modified: 2022-04-27
MMDB ID: 214913 PDB ID: 7Z0P
View in iCn3D Similar Structures PubChem Compound

3. Locked Wuhan SARS-CoV2 Prefusion Spike ectodomain with lipid bound[VIRAL PROTEIN]
Taxonomy: Severe acute respiratory syndrome coronavirus 2
Proteins: 3 Chemicals: 69 modified: 2022-05-04
MMDB ID: 215228 PDB ID: 7Z3Z
View in iCn3D Similar Structures PubMed PubChem Compound



7L7F: Cryo-EM structure of human ACE2 receptor bound to protein encoded by vaccine candidate BNT162b1

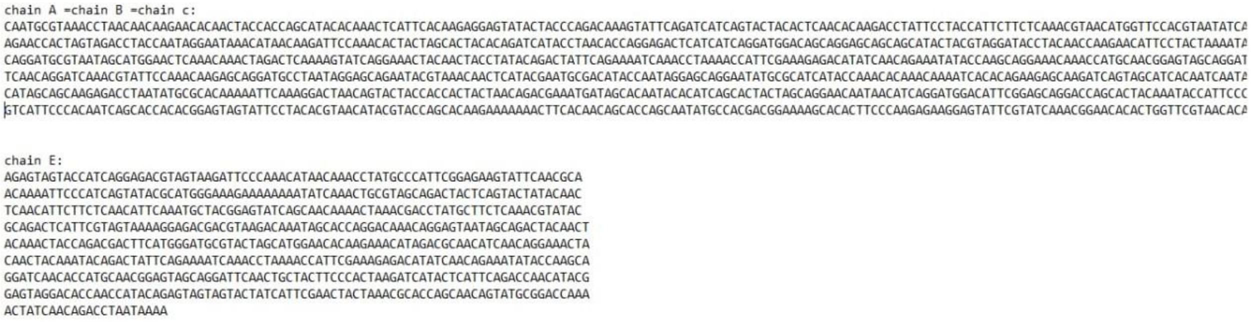
Format: ASN.1 (Cn3D)

Data Set: Single 3D structure

Figure 19

This figure shows many chains with spike protein.

And we did the same thing for Omicron and also RNN model predicted it correctly(+ve).

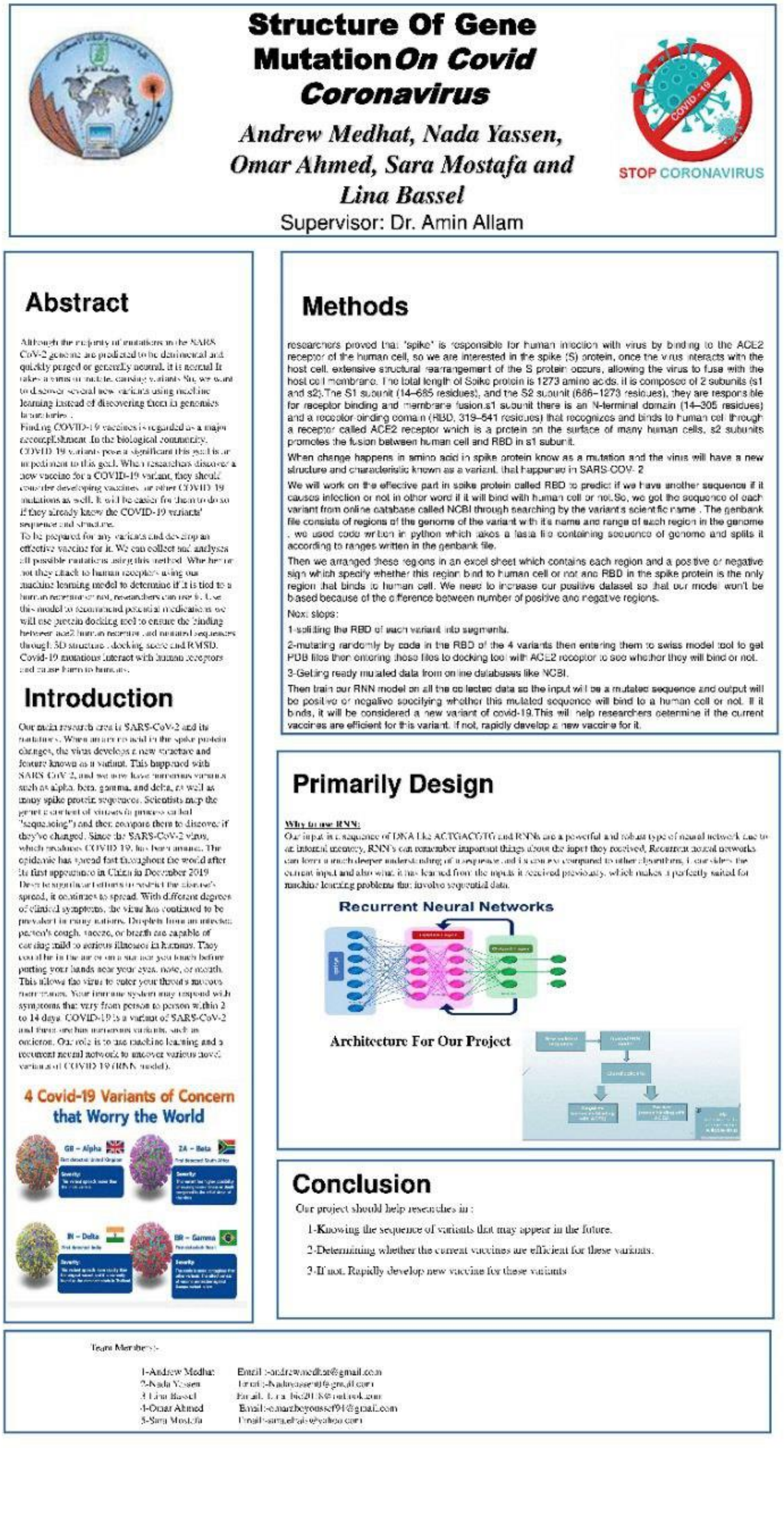


```
chain A =chain B =chain c:
CAATGCGTAAACCTAACAACAAGAACACAACTACCACCAGCATACACAAACTCATTCACAAGAGGAGTATACTACCCAGACAAAGTATTCAGATCATCAGTACTACACTCAACACAAGACCTATTCCTACCATTCTTCTCAAACGTAACATGGTTCCACGTAATATCA
AGAACCACTAGTAGACCTACCAATAGGAATAAACATAACAAGATTCCAAACACTACTAGCACTACACAGATCATACCTAACACCAGGAGACTCATCATCAGGATGGACAGCAGGAGCAGCAGCATACTACGTAGGATACCTACAACCAAGAACATTCCTACTAAAATA
CAGGATGCGTAATAGCATGGAACTCAAACAAACTAGACTCAAAAGTATCAGGAAACTACAACTACCTATACAGACTATTCAGAAAATCAAACCTAAAACCATTCGAAAGAGACATATCAACAGAAATATACCAAGCAGGAAACAAACCATGCAACGGAGTAGCAGGAT
TCAACAGGATCAAACGTATTCCAAACAAGAGCAGGATGCCTAATAGGAGCAGAATACGTAAACAACTCATACGAATGCGACATACCAATAGGAGCAGGAATATGCGCATCATACCAAACACAAACAAAATCACACAGAAGAGCAAGATCAGTAGCATCACAATCAATA
CATAGCAGCAAGAGACCTAATATGCGCACAAAAATTCAAAGGACTAACAGTACTACCACCACTACTAACAGACGAAATGATAGCACAATACACATCAGCACTACTAGCAGGAACAATAACATCAGGATGGACATTCGGAGCAGGACCAGCACTACAAATACCATTCCC
GTCATTCCCACAATCAGCACCACACGGAGTAGTATTCCTACACGTAACATACGTACCAGCACAAGAAAAAAACTTCACAACAGCACCAGCAATATGCCACGACGGAAAAGCACACTTCCCAAGAGAAGGAGTATTCGTATCAAACGGAACACACTGGTTCGTAACACA


chain E:
AGAGTAGTACCATCAGGAGACGTAGTAAGATTCCCAAACATAACAAACCTATGCCCATTCGGAGAAGTATTCAACGCA
ACAAAATTCCCATCAGTATACGCATGGGAAAGAAAAAAAATATCAAACTGCGTAGCAGACTACTCAGTACTATACAAC
TCAACATTCTTCTCAACATTCAAATGCTACGGAGTATCAGCAACAAAACTAAACGACCTATGCTTCTCAAACGTATAC
GCAGACTCATTCGTAGTAAAAGGAGACGACGTAAGACAAATAGCACCAGGACAAACAGGAGTAATAGCAGACTACAACT
ACAAACTACCAGACGACTTCATGGGATGCGTACTAGCATGGAACACAAGAAACATAGACGCAACATCAACAGGAAACTA
CAACTACAAATACAGACTATTCAGAAAATCAAACCTAAAACCATTCGAAAGAGACATATCAACAGAAATATACCAAGCA
GGATCAACACCATGCAACGGAGTAGCAGGATTCAACTGCTACTTCCCACTAAGATCATACTCATTCAGACCAACATACG
GAGTAGGACACCAACCATACAGAGTAGTAGTACTATCATTCGAACTACTAAACGCACCAGCAACAGTATGCGGACCAAA
ACTATCAACAGACCTAATAAAA
```

Figure 20

This figure shows the DNA sequence of Omicron

Poster



Structure Of Gene Mutation *On Covid Coronavirus*

Andrew Medhat, Nada Yassen, Omar Ahmed, Sara Mostafa and Lina Bassel

Supervisor: Dr. Amin Allam

STOP CORONAVIRUS

Abstract

Although the majority of mutations in the SARS CoV-2 genome are predicted to be detrimental and quickly purged or generally neutral. it is normal It takes a virus to mutate causing variants So, we want to discover several new variants using machine learning instead of discovering them in genomics laboratories .
Finding COVID-19 vaccines is regarded as a major accomplishment .In the biological community. COVID-19 variants pose a significant this goal is an impediment to this goal. When researchers discover a new vaccine for a COVID-19 variant, they should consider developing vaccines for other COVID-19 mutations as well. It will be easier for them to do so if they already know the COVID-19 variants' sequence and structure.
To be prepared for any variants and develop an effective vaccine for it. We can collect and analyses all possible mutations using this method. Whether or not they attach to human receptors using our machine learning model to determine if it is tied to a human receptor or not, researchers can use it. Use this model to recommend potential medications we will use protein docking tool to ensure the binding between ace2 human receptor and mutated sequences through 3D structure , docking score and RMSD. Covid-19 mutations interact with human receptors and cause harm to humans.

Introduction

Our main research area is SARS-CoV-2 and its mutations. When an amino acid in the spike protein changes, the virus develops a new structure and feature known as a variant. This happened with SARS-CoV-2, and we now have numerous variants such as alpha, beta, gamma, and delta, as well as many spike protein sequences. Scientists map the genetic content of viruses (a process called "sequencing") and then compare them to discover if they've changed. Since the SARS-CoV-2 virus, which produces COVID-19, has been around. The epidemic has spread fast throughout the world after its first appearance in China in December 2019 Despite significant efforts to restrict the disease's spread, it continues to spread. With different degrees of clinical symptoms, the virus has continued to be prevalent in many nations. Droplets from an infected person's cough, sneeze, or breath are capable of causing mild to serious illnesses in humans. They could be in the air or on a surface you touch before putting your hands near your eyes, nose, or mouth. This allows the virus to enter your throat's mucous membranes. Your immune system may respond with symptoms that vary from person to person within 2 to 14 days. COVID-19 is a variant of SARS-CoV-2 and there are has numerous variants, such as omicron. Our role is to use machine learning and a recurrent neural network to uncover various novel variants of COVID-19 (RNN model).

4 Covid-19 Variants of Concern that Worry the World

Methods

researchers proved that 'spike' is responsible for human infection with virus by binding to the ACE2 receptor of the human cell, so we are interested in the spike (S) protein, once the virus interacts with the host cell. extensive structural rearrangement of the S protein occurs, allowing the virus to fuse with the host cell membrane. The total length of Spike protein is 1273 amino acids. it is composed of 2 subunits (s1 and s2).The S1 subunit (14–685 residues), and the S2 subunit (686–1273 residues), they are responsible for receptor binding and membrane fusion.s1 subunit there is an N-terminal domain (14–305 residues) and a receptor-binding domain (RBD, 319–541 residues) that recognizes and binds to human cell through a receptor called ACE2 receptor which is a protein on the surface of many human cells, s2 subunits promotes the fusion between human cell and RBD in s1 subunit.

When change happens in amino acid in spike protein know as a mutation and the virus will have a new structure and characteristic known as a variant. that happened in SARS-COV- 2

We will work on the effective part in spike protein called RBD to predict if we have another sequence if it causes infection or not in other word if it will bind with human cell or not.So, we got the sequence of each variant from online database called NCBI through searching by the variant's scientific name . The genbank file consists of regions of the genome of the variant with it's name and range of each region in the genome , we used code written in python which takes a fasta file containing sequence of genome and splits it according to ranges written in the genbank file.

Then we arranged these regions in an excel sheet which contains each region and a positive or negative sign which specify whether this region bind to human cell or not and RBD in the spike protein is the only region that binds to human cell. We need to increase our positive dataset so that our model won't be biased because of the difference between number of positive and negative regions.

Next steps:

1-splitting the RBD of each variant into segments.

2-mutating randomly by code in the RBD of the 4 variants then entering them to swiss model tool to get PDB files then entering those files to docking tool with ACE2 receptor to see whether they will bind or not.

3-Getting ready mutated data from online databases like NCBI.

Then train our RNN model on all the collected data so the input will be a mutated sequence and output will be positive or negative specifying whether this mutated sequence will bind to a human cell or not. If it binds, it will be considered a new variant of covid-19.This will help researchers determine if the current vaccines are efficient for this variant. If not, rapidly develop a new vaccine for it.

Primarily Design

Why to use RNN:
Our input is a sequence of DNA like ACTGACGTG and RNNs are a powerful and robust type of neural network due to an internal memory, RNN's can remember important things about the input they received, Recurrent neural networks can form a much deeper understanding of a sequence and its context compared to other algorithms, it considers the current input and also what it has learned from the inputs it received previously. which makes it perfectly suited for machine learning problems that involve sequential data.

Recurrent Neural Networks

Architecture For Our Project

Conclusion

Our project should help researches in :

1-Knowing the sequence of variants that may appear in the future.

2-Determining whether the current vaccines are efficient for these variants.

3-If not. Rapidly develop new vaccine for these variants

Team Members:-

1-Andrew Medhat Email :-andrewmedhat@gmail.com
2-Nada Yassen Email:-Nadayassen@gmail.com
3-Lina Bassel Email:-Lina [illegible]2018@outlook.com
4-Omar Ahmed Email:-omarhebyoussef94@gmail.com
5-Sara Mostafa Email:-sara[illegible]@yahoo.com