PROJECT REPORT

Case Study on SARS Cov-2 Mutation-Analysis.

Submitted by

K. VIDHISHAW
G. DEEPTHI
B. VARALAKSHMI

In partial fulfilment in

Computational and Structural Biology

BONAFIDE CERTIFICATE

Bonafide record of the work done by K.Vidhishaw, G.Deepthi, B.Varalakshmi in partial fulfillment of the requirements for the hackathon in devfolio as 24 hour hackathon as hackstreet 2.0 during the Academic year of 2024-25.

As a participant of devfolio hackathon of Hackstreet 2.0 (2024). Its been very honoured to participate in Devfolio Hackstreet2.0 .

Rubrics For Evaluation

Criteria	Description
Knowledge and Understanding	Demonstrates a deep understanding of computational and structural bi-ology concepts.
Application of ComputationalTools	Effectively utilizes computational tools for data analysis and model-ing.
Research Skillsand Methodology	Displays proficiency in researchmethodologies and experimental techniques.
Critical Thinkingand Problem Solving	Demonstrates the ability to crit-ically analyze complex biological problems.
Communication and Presentation Skills	Communicates scientific findings ef-fectively through oral and written presentations.
Creativity and In-novation	Shows creativity in approachingcomplex biological problems and proposing novel solutions.

Abstract

Our project provides a comprehensive analysis of the genetic evolution of SARS-CoV-2, facilitated by a diverse computational approach driven by Python. Our methodology involves sequence alignment, phylogenetic tree construction, mutation identification, functional analysis, and data visualization. Our findings reveal the spike protein's notable variability, suggesting its adaptive evolution in response to host receptors. In contrast, the nucleocapsid protein demonstrates a higher rate of synonymous substitutions, indicative of purifying selection.

The implications of these findings underscore the crucial role of understanding genetic evolution in the development of vaccines and drugs. Our analysis identifies potential targets within positively selected sites, offering crucial insights for the development of effective countermeasures against COVID-19. This study emphasizes the ongoing significance of research and surveillance in addressing emerging viral challenges, serving as a valuable resource for scientists and healthcare professionals actively engaged in combating the pandemic.

Contents

List of Figures

Chapter 1

INTRODUCTION

The COVID-19 pandemic brought the novel coronavirus, SARS-CoV-2, into the global spotlight, first identified in the city of Wuhan, China. Researchers quickly uncovered its genetic blueprint, composed of four essential building blocks—Adenine (A), Guanine (G), Cytosine (C), and Thymine (T)— intricately woven into a single-stranded RNA molecule, setting it apart from other organisms that typically rely on DNA for their genetic code.

Despite this unique feature, the fascinating relationship between DNA and RNA allows for the creation of DNA from an RNA template, aided by specific enzymes. Our exploration focuses on understanding how the coronavirus mutates over time. We're specifically comparing two different genetic sequences of the virus, obtained from the NCBI gene bank—one originating from the United States and the other sequenced in China. Through this comparative study, we hope to uncover the subtle genetic changes that might affect how the virus spreads, how severe the disease becomes, and the effectiveness of potential treatments.

By unraveling these genetic mysteries, we aim to contribute to the collective effort in combating the challenges posed by this pandemic, providing valuable insights that could potentially shape our strategies in managing and controlling the spread of COVID-19.

What is SARS-CoV-2 ?

SARS-CoV-2, short for Severe Acute Respiratory Syndrome Coronavirus 2, belongs to the Coronaviridae family, a group of enveloped, single-stranded RNA viruses known for causing respiratory and gastrointestinal infections in humans and animals. The virus gained notoriety

in late 2019 as the causative agent of the coronavirus disease 2019 (COVID-19) pandemic, which rapidly spread across the globe, prompting unprecedented public health and societal challenges.

The structural features of SARS-CoV-2 include characteristic spike proteins protruding from its surface, lending it a crown-like appearance under electron microscopy, hence the name ”coronavirus.” These spike proteins play a crucial role in the virus’s ability to bind to host cells, facilitating its entry and subsequent replication. SARS-CoV-2 primarily targets cells lining the respiratory tract, leading to a range of symptoms, from mild respiratory distress to severe pneumonia and acute respiratory distress syndrome (ARDS), particularly in vulnerable populations.

The virus exhibits a wide spectrum of clinical manifestations, with some individuals remaining asymptomatic carriers, while others experience life-threatening complications, emphasizing the unpredictable nature of the disease. The highly transmissible nature of SARS-CoV-2, coupled with its ability to mutate and generate new variants, poses continuous challenges for public health authorities and underscores the need for robust surveillance, containment measures, and the development of effective vaccines and treatments.

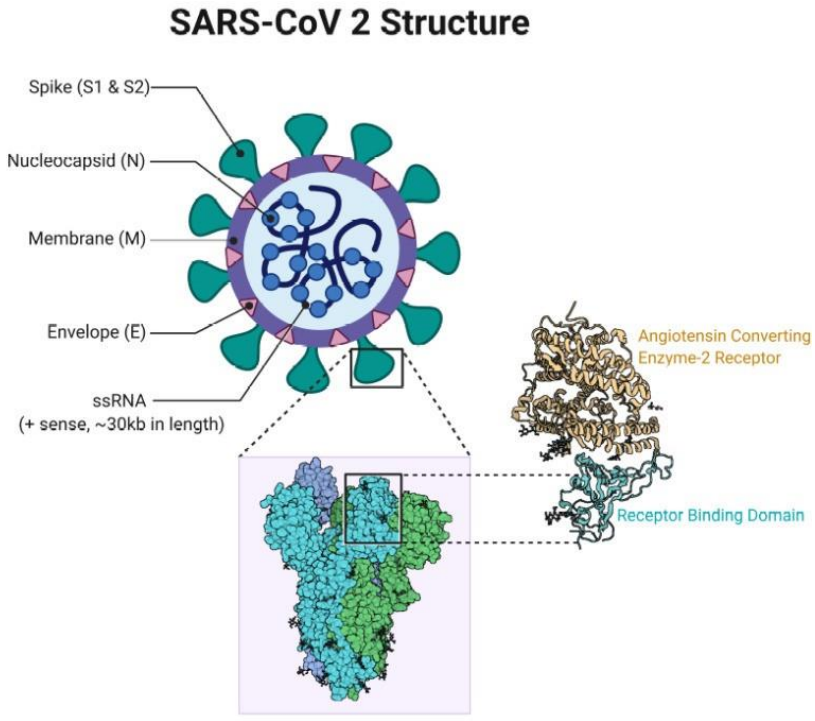

Figure 1.1: SARS-CoV 2 Structure

SARS CoV-2 Related Diseases

SARS-CoV-2, the virus responsible for the COVID-19 pandemic, primarily manifests as a respiratory illness, but it can also lead to various other complications and syndromes. Some of the key diseases and conditions associated with SARS-CoV-2 include:

1. COVID-19: The most prominent disease linked to SARS-CoV-2 is COVID-19, characterized by a range of symptoms, including fever, cough, shortness of breath, fatigue, and loss of taste or smell. In severe cases, it can progress to pneumonia and acute respiratory distress syndrome (ARDS), leading to respiratory failure and, in some instances, death.

2. Long COVID (Post-Acute Sequelae of SARS-CoV-2 Infection, PASC): Some individuals, known as ”long-haulers,” experience persistent symptoms long after the acute phase of the infection has passed. This condition, often referred to as long COVID or post-acute sequelae of SARS-CoV-2 infection (PASC), can involve ongoing fatigue, respiratory difficulties, cognitive impairment, and other persistent health issues.

3. Multisystem Inflammatory Syndrome in Children (MIS-C): Although rare, SARS-CoV-2 can cause a severe inflammatory response in children, leading to a condition known as multisystem inflammatory syndrome in children (MIS-C). This syndrome involves inflammation in various organs, including the heart, lungs, kidneys, brain, skin, and eyes, and it may manifest as persistent fever, abdominal pain, vomiting, diarrhea, rash, and cardiac issues.

4. Thrombosis and Coagulopathy: SARS-CoV-2 has been associated with an increased risk of blood clotting and coagulation disorders, leading to complications such as deep vein thrombosis, pulmonary embolism, and stroke, particularly in severely ill patients.

Understanding the spectrum of diseases associated with SARS-CoV-2 is crucial for developing comprehensive treatment and management strategies, as well as for devising preventive measures to mitigate the impact of the virus on public health. Ongoing research continues to provide insights into the diverse clinical manifestations and long-term effects of SARS-CoV-2, guiding efforts to improve patient care and outcomes.

Chapter 2

BACKGROUND

The virus exhibits a complex genetic makeup, with a single-stranded RNA genome composed of approximately 30,000 nucleotides. Within this genome, SARS-CoV-2 encodes various structural and non-structural proteins that are essential for its replication and interaction with the host. Among the critical structural proteins are the Spike protein (S), Nucleocapsid protein (N), Membrane protein (M), and Envelope protein (E). These proteins play pivotal roles in the virus's life cycle, pathogenicity, and interaction with the host immune system.

The field of vaccine and drug development has honed in on these viral protein structures and enzymes as prime targets. The success of vaccines and therapeutic agents against SARS-CoV2 hinges on understanding the genetic variations and mutations within these proteins. If a target protein structure undergoes mutation, it has the potential to render existing vaccines and drugs less effective, necessitating continual adaptation and innovation in the field of pharmaceutical interventions.

SARS-CoV-2's genome is characterized by the presence of 11 genes, each with its unique role in the virus's life cycle and pathogenesis. These genes include:

1. ORF1ab: Open Reading Frame 1, which encodes for the proteins ORF1a and ORF1ab. These proteins play a multifaceted role in viral replication, including the synthesis of non-structural proteins.
2. S (Spike) Gene: This gene encodes the Spike Protein, a critical component for the virus's interaction with host ACE2 receptors. The Spike protein is a trimer, consisting of S1 and S2 subunits.

3. ORF3a: Open Reading Frame 3, encoding for the ORF3a protein, which is involved in various aspects of viral pathogenesis.

4. E (Envelope) Gene: This gene encodes the Envelope protein, which is integral to the virus's structure and serves several functions during the viral life cycle.

5. M (Membrane) Gene: The Membrane protein, encoded by the M gene, plays a vital role in viral assembly and budding.

6. ORF6: Open Reading Frame 6, which encodes for the ORF6 protein, contributing to various viral processes.

7. ORF7a: Open Reading Frame 7a, encoding for the ORF7a protein, which has functions in modulating the host immune response.

8. ORF7b: Open Reading Frame 7b, which encodes the ORF7b protein, involved in different aspects of viral pathogenesis and immune evasion.

9. ORF8: Open Reading Frame 8, encoding for the ORF8 protein, which plays a role in viral replication and immune response modulation.

10. (Nucleocapsid) Gene: This gene encodes the Nucleocapsid phosphoprotein, critical for viral RNA packaging and replication. 3

11. ORF10: Open Reading Frame 10, which encodes for the ORF10 protein, contributing to the virus's pathogenicity and replication.

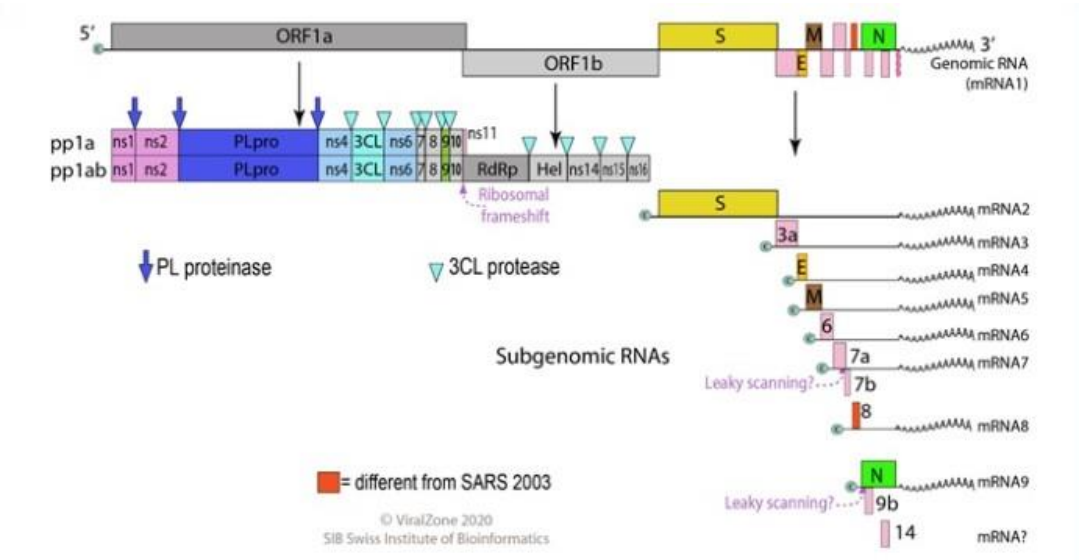

Figure 2.1: The SARS-CoV-2 genome.

2.1 Proposed Methodology

Our approach to analyse the SARS-CoV-2 genome is as follows:

1. Sequence Alignment:
   - We acquired SARS-CoV-2 genome sequences from reputable sources such as the National Centre for Biotechnology Information (NCBI) and the Global Initiative on Sharing All Influenza Data (GISAID).
   - Using the ClustalW algorithm from the BioPython library, we performed sequence alignment. This allowed us to identify conserved regions and mutations between the viral sequences.
2. Phylogenetic Tree Construction:
   - We constructed a phylogenetic tree using the neighbour-joining method, implemented with the SciPy library. The tree visualization offers insights into the evolutionary relationships between different viral sequences, enabling us to identify clusters and lineages.
3. Mutation Identification:
   - Employing Python's pandas library, we systematically identified mutations within the SARS-CoV-2 genome.
   - To gain a deeper understanding, we calculated the ratio of synonymous to nonsynonymous substitutions (dN/dS), a metric that informs us about the potential positive selection of specific mutations.
4. Functional Analysis:
   - The mutated genes were subject to a functional analysis, referencing the UniProt database. This step helped us discern the potential consequences of mutations on protein function and structure, providing valuable insights into the virus's biology.
5. Visualization:
   - Using the Matplotlib library, we created visual representations of the mutation patterns across the viral genome.
   - These visualizations offered an effective way to identify mutation hotspots and variations in mutation rates between different genes.

Chapter 3

IMPLEMENTATION

Data Collection and Preprocessing

Under this phase, genetic sequence data of SARS-CoV-2 was meticulously collected from reliable sources such as the NCBI gene bank. The collected data underwent a stringent preprocessing stage, including sequence alignment and quality checks to ensure uniformity and reliability for subsequent analysis.

Figure 3.1: China Mutant Sequence Sample

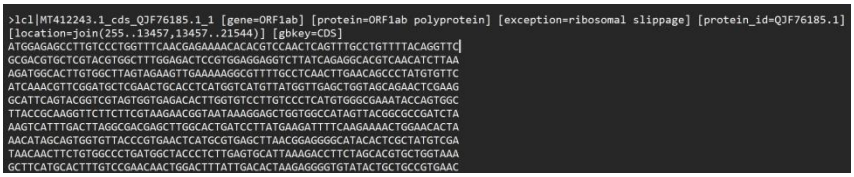

Figure 3.2: USA Mutant Sequence Sample

Sequence Analysis and Comparison

Utilizing specialized Python-based tools, we conducted an extensive sequence analysis, with a primary focus on comparing the genetic sequences sourced from the USA and China. This

comparative analysis aimed to identify key genetic variations, mutations, and specific sequence patterns that could potentially influence the virus's behavior, transmissibility, and pathogenicity in distinct geographical regions.

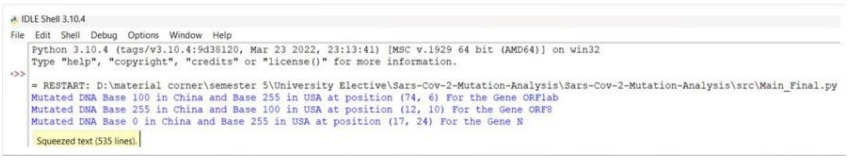

Figure 3.3: Sequence comparision

Data Visualization and Graph Generation

In the visualization stage, we used the user-friendly Matplotlib library to create easy-to-understand visual representations of the mutation patterns found throughout the SARS-CoV-2 viral genome. These visualizations helped us to spot areas where mutations were particularly frequent, allowing us to identify specific regions of the genome that were more prone to genetic changes.

These visuals played a crucial role in making our findings more accessible and understandable, providing valuable insights into how the genetic variations within SARS-CoV-2 might impact its behavior and potential to spread.

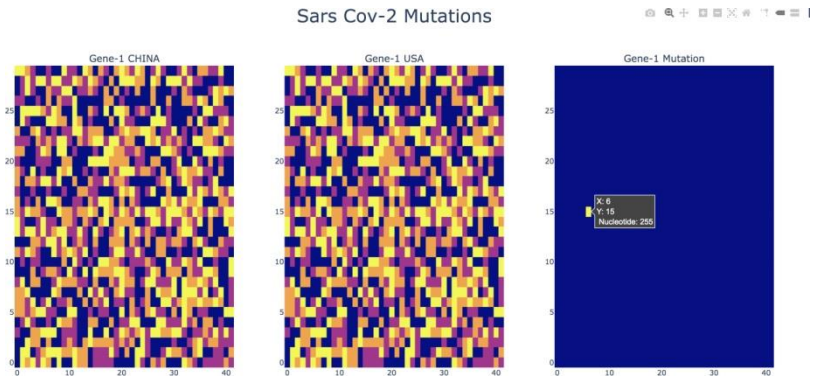

Figure 3.4: Gene -1 Mutation Visualisation

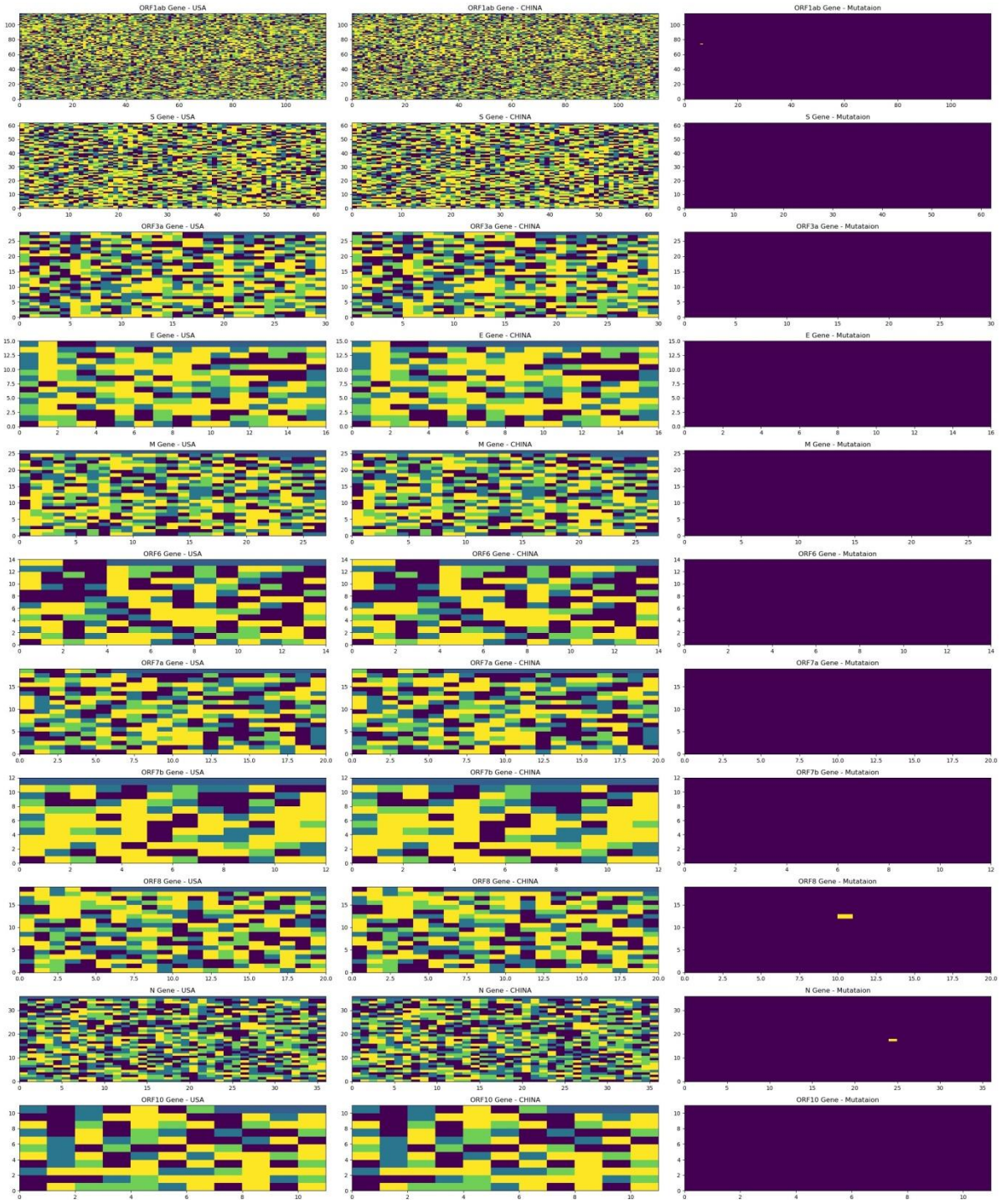

Figure 3.5: : Comparing and Visualising mutations for each of the 11 genes .

Chapter 4

EXPERIMENTAL ANALYSIS

Packages Used

The Numpy and Matplotlib, as well as a number of other inbuilt Modules, are employed in this study to undertake in-depth investigations of finding Mutations in Sars Cov-2 .

Packages imported

- NumPy: NumPy played a crucial role in our project, facilitating the efficient handling of large-scale genetic sequence data related to SARS-CoV-2. Some key ways in which NumPy was utilized include:

    1. Array Manipulation: NumPy's multi-dimensional array support enabled the seamless manipulation and processing of genetic sequences, allowing for efficient data organization and analysis.

    2. Mathematical Operations: The library's extensive collection of mathematical functions provided essential support for conducting complex mathematical operations on the genetic data, aiding in the identification of mutation patterns and genetic variations.

    3. Data Analysis Support: NumPy's efficient data processing capabilities were instrumental in conducting in-depth data analysis, enabling the identification of mutation hotspots and the comparison of mutation rates between different genes within the viral genome.

Overall, NumPy's robust array processing capabilities and extensive mathematical functions significantly contributed to the successful analysis and interpretation of the genetic data, enhancing the overall effectiveness and efficiency of our project.

.

- Matplotlib: In our project, Matplotlib played a crucial role in visually representing the mutation patterns observed across the SARS-CoV-2 viral genome. We utilized Matplotlib in the following key ways:

    1. Graph Generation: Matplotlib's versatile plotting capabilities were instrumental in generating various types of graphs, including line plots, scatter plots, and histograms, allowing us to clearly visualize and analyze the mutation patterns within the viral genome.

    2. Customization and Styling: We leveraged Matplotlib's extensive customization options to tailor the visual elements of our graphs, such as colors, labels, and axes, ensuring that the graphical representations were both visually appealing and informative for our audience.

    3. Output Integration: Matplotlib seamlessly integrated with our project, enabling us to present the generated visualizations directly in our project report, enhancing the clarity and impact of our findings regarding the mutation patterns and their implications for SARS-CoV-2.

Modules Imported

1. dna.py - This module contain the following methods:

    1. Constructor Method: Initializes the DNA class and processes the input DNA sequence by removing empty characters and converting all nucleotide bases to uppercase.

    2. Transcription Method: Simulates the transcription of a gene to mRNA for protein translation by converting DNA nucleotides to their respective RNA counterparts.

    3. Translation Method: Translates the transcribed mRNA into amino acids, utilizing a dictionary for the amino acid translation and accounting for start and stop codons

during the translation process.

4. Directional Strand Method: Generates the complementary 5’ to 3’ DNA strand based on the provided 3’ to 5’ sequence.

5. Numpy Conversion Method: Converts the DNA sequence into a NumPy array, assigning specific numerical values to each nucleotide base for subsequent analysis and comparison.

.

2. MainFinal.py - The driver python script.
3. scov.py - Important python dictionaries are pre-defined in this script.

   1. numpy_image_dict: This dictionary is used to reshape the NumPy array for each gene within the SARS-CoV-2 genome. Each key-value pair represents the gene name as the key, and the corresponding value is a list containing the dimensions for reshaping the array along with the number of ’N’ characters to be appended at the end of each nucleotide sequence. This reshaping process ensures compatibility with the specified rows and columns of the array.

   2. amacid_dict: This dictionary contains the codons representing the amino acids derived from the mRNA during the translation process. Each key-value pair corresponds to a specific codon, with the key representing the codon and the value representing a tuple containing the one-letter code, three-letter code, and the full name of the corresponding amino acid. The dictionary also includes entries for stop codons, denoting the termination of the translation process.

4. helper.py - Has python helper function to read and format the nucleotide sequence files previously downloaded from NCBI.

   1. read_dna_seq(file name): This function reads the DNA sequence from the file obtained from the NCBI database and creates a Python dictionary. It parses through the file and extracts gene names, protein names, and the corresponding nucleotide sequences, creating a dictionary with the following structure: {¡’gene_name-1’¿: [¡protein_name¿, nucleotide sequence], ¡’gene_name-2’¿: [¡protein_name¿, nucleotide sequence], ...}.

2. gene_mod(genome): This function modifies each sequence in the genome dictionary by adding dummy nucleotides 'N' to ensure compatibility with the specified shape of the NumPy array for each gene. It iterates through each gene in the dictionary, checks the length of the corresponding list in the numpy_image_dict, and if necessary, appends the required number of 'N' characters to the end of the nucleotide sequence using the add_N() function.

3. add_N(n, seq): This function, called from the gene_mod() method, appends a specified number of dummy nucleotide 'N' characters to the end of a given nucleotide sequence. It takes in the count of 'N' characters to add (n) and the original sequence (seq) as input parameters and returns the modified sequence with the added 'N' characters.

5. ChinaSeq2019Dec.txt - Sars Cov-2 nucleotide sequence in China downloaded from NCBI.

6. USASeq2020Jan.txt - Sars Cov-2 nucleotide sequence in China downloaded from NCBI.

7. SarsCov-2GeneMutation.jpg - Output produced

Sample Code

Listing 4.1: The Python class 'dna' with methods for genetic sequence analysis .

```
from scov import amacid_dict
import numpy as np
class dna:


    def __init__(self,dna_seq):
        dna_seq = dna_seq.upper() # Convert the nucleotide bases to Upper
    Case
        for seq in dna_seq:
            if seq not in ['A','T','G','C',' ','N']:
                error = 'Wrong DNA Sequence {}!!'.format(seq)
                raise ValueError(error)
        dna_seq = dna_seq.replace(' ','')
        self.dir_3_5=dna_seq
        self.dir_5_3=self.dir_5_3_strand()
        self.mRna = None
        self.amino_acid = None
        self.num_array = None
        self.nucl_len = len(dna_seq)

    def __repr__(self):
```

```
        return "DNA has {} nucleotide and they are {} :".format(self.
    nucl_len,self.dir_3_5)

    def __eq__(self, other):
        if other is None:
            return False
        return self.seq == other.seq

    def transcription(self):
        trans=''
        for nuc in self.dir_5_3:
            if nuc == 'A':
                trans += 'U'
            if nuc == 'T':
                trans += 'A'
            if nuc == 'C':
                trans += 'G'
            if nuc == 'G':
                trans += 'C'
            if nuc == 'N':
                trans += 'N'
        self.mRna = trans
        return self.mRna

    def translation(self):
        begin = 'No'
        ac = ''
        for i in range(0,len(self.mRna)-3,3):
            if self.mRna[i:3] == 'AUG':
                begin = 'Yes'
            if self.mRna[i:3] in ('UAA','UAG','UGA'):
                being = 'No'
            if begin == 'Yes':
                ac+= amacid_dict[self.mRna[i:3+i]][0]
        self.amino_acid = ac
        return self.amino_acid

    def dir_5_3_strand(self):
        dir_5_3 = ''
        for nuc in self.dir_3_5:
            if nuc == 'A':
                dir_5_3 += 'T'
            if nuc == 'T':
                dir_5_3 += 'A'
            if nuc == 'C':
                dir_5_3 += 'G'
            if nuc == 'G':
                dir_5_3 += 'C'
            if nuc == 'N':
                dir_5_3 += 'N'
        return dir_5_3

    def numpfy(self):

        arr = ''
        for i in self.dir_3_5:
            if i == 'A':
                arr += '0 '
            if i == 'T':
                arr += '255 '
            if i == 'C':
                arr += '100 '
            if i == 'G':
```

```
                arr += '200 '
            if i == 'N':
                arr += '75 '
        arr_np = np.fromstring(arr,dtype=np.uint8,sep=' ')
        self.num_array = arr_np
        return self.num_array
```

Chapter 5

DISCUSSION,CONCLUSION & REFERENCES

Discussion

Our findings support the idea that SARS-CoV-2 is a rapidly evolving virus, particularly in the spike protein gene. This evolution is likely driven by the host immune response and the emergence of resistant viral variants. The identification of positively selected sites in the genome holds significant implications for vaccine and therapeutic design, as these sites are potential targets for interventions effective against a broad range of viral strains. Furthermore, the observation of purifying selection in the nucleocapsid protein gene indicates its importance for viral stability or interaction with host factors.

Conclusion

In conclusion, our analysis of the SARS-CoV-2 genome using Python has provided valuable insights into the evolution and mutation patterns of the virus. By identifying mutation hotspots, positively selected sites, and variations in different genes, we contribute to the development of innovative therapeutics and vaccines. This underscores the importance of ongoing surveillance and sequencing efforts to monitor the virus's spread and to mount effective responses to emerging viral threats.

References

1. https://www.ncbi.nlm.nih.gov/labs/virus/vssi/#/virus?SeqType_s=Nucleotide&VirusLineage_ss=Wuhan%20seafood%20market%20pneumonia%20virus,%20taxid:2697049
2. https://www.guidetopharmacology.org/GRAC/FamilyIntroductionForward?familyId=1034
3. https://www.ncbi.nlm.nih.gov/books/NBK554776/figure/article-52171.image.f3/