Capstone Project:

Taxonomic Classification of DNA and RNA Viruses Using Machine Learning

Individual Report for the Capstone Project

Author:
Pang-Kuo Lo

Professor:
Dr. Rabih Jabbour

Course:
Capstone Project (BIOT 670I)

Master Program:
Biotechnology specialized in Bioinformatics

University of Maryland Global Campus (UMGC)

Date:
October 8, 2023

Project Title: Taxonomic Classification of DNA and RNA Viruses

Executive Summary

The field of virology is at the forefront of understanding and combatting viral infections, but traditional methods of viral classification are challenged by the dynamic and diverse nature of viruses. This project addresses the need for innovative artificial intelligence/machine learning (AI/ML) methods alongside Multiple Sequence Alignment (MSA) algorithms to classify viruses more accurately and efficiently.

The recent explosion in available viral genomes has led to a need for adaptable tools to classify a wide range of virus strains across diverse virus families. The project aims to create an AI/ML-driven taxonomy classification tool to identify viruses, especially emerging strains. To ensure reliability, a diverse dataset spanning major virus families and genera will be used. A comparative analysis against established bioinformatics tools will assess the tool's accuracy, efficiency, and adaptability. Stringent quality control measures will minimize errors. The outcome of this project is a sophisticated AI/ML-driven taxonomy classification tool that enhances our understanding of viral diversity, aiding researchers and healthcare professionals in responding to emerging viral threats effectively.

The project's research has advanced viral genus classification significantly, achieving accuracy metrics surpassing 99% with supervised machine learning models. Unsupervised clustering techniques revealed insights into viral protein sequences' structural aspects. The flexibility and the usability of the ML-based tools surpass existing bioinformatics tools. The project's findings hold great promise for virology, epidemiology, and the scientific community, emphasizing the need for continued research and innovation in viral taxonomy classification.

Introduction

The field of virology has witnessed remarkable progress in understanding and combating viral infections. From the pioneering work of scientists like Dmitri Ivanovsky and Martinus Beijerinck in the late 19th century (Lecoq, 2001) to the modern era of genomics and molecular biology, our knowledge of viruses and their impact on human health has grown exponentially. Central to this progress has been the classification of viruses, providing the cornerstone for comprehending these microscopic entities. While traditional methods of viral classification, such as morphological, serological, or biochemical characteristics, have been effective in organizing the viral world, they may not be sufficient considering the dynamic and diverse nature of viruses.

It is important to note that the need for innovative artificial intelligence/machine learning (AI/ML) methods alongside effective Multiple Sequence Alignment (MSA) algorithms in the field of viral classification stems from the unique challenges posed by viral diversity (Ao et al., 2022). While MSA algorithms have proven their effectiveness in aligning and comparing viral sequences, viruses continually evolve, mutate, and diversify. This presents a substantial challenge in accurately classifying them based solely on their genetic information. The limitations of MSA algorithms become evident when dealing with rapidly evolving viruses or those with highly diverse genomes.

In this context, AI and ML algorithms offer a complementary approach. They can handle vast datasets, identify subtle patterns, and adapt to the ever-changing viral landscape (Afify &

Zanaty, 2021; Bzhalava et al., 2018; Randhawa et al., 2020; Remita et al., 2017; Tang et al., 2022). The dynamic nature of viruses and the sheer volume of data available require innovative solutions to classify viruses accurately and efficiently, paving the way for improved diagnostics, treatments, and prevention strategies. This project is aimed at highlighting the compelling need for AI/ML methods alongside MSA algorithms, which can unlock new insights and enhance capacity to combat viral infections in this rapidly evolving field.

Project Background:

The recent strides in cloning and sequencing technologies have sparked a profound transformation in the realm of virology. This transformation is marked by an unprecedented proliferation of available viral genomes. These genetic blueprints, cataloging the intricate details of various viruses, represent a treasure trove of information that carries profound implications for our understanding of viral diversity, taxonomy, and disease mechanisms.

The existing methods for classifying and annotating viral genomes have primarily been designed for well-studied virus families, leaving a critical gap in our ability to efficiently and accurately categorize the ever-expanding repertoire of newly sequenced virus strains across diverse virus families. This gap presents a pressing need for the development of more adaptable and streamlined tools capable of accommodating the remarkable diversity of viruses.

By addressing this need, these innovative tools promise to enhance the quality and depth of viral comparative genomic investigations. They will empower researchers to explore genetic variations, taxonomic nuances, and uncover insights into the underlying mechanisms of diseases caused by a vast array of viruses. The integration of adaptable and accurate classification techniques into the field of virology is poised to revolutionize our understanding of viruses and their impact on human and environmental health.

Project Objectives and Description:

Overview:

Our project seeks to create an innovative AI/ML-driven taxonomy classification tool with a primary focus on identifying viruses, including emerging viral strains. In the face of rapidly evolving viral threats, the development of a reliable and adaptable tool is of paramount importance to advance our understanding of viral diversity and pathogenicity. This project combines cutting-edge technologies with comprehensive training and rigorous evaluation to meet this critical need.

Dataset and Taxonomy:

To ensure the robust performance of our classification tool, we will curate a diverse and representative dataset. This dataset will encompass viruses from four major families, spanning seven subfamilies and twenty genera. This broad selection is designed to provide the tool with the necessary foundation to make predictions at the viral genus level accurately. The inclusion of emerging and less-studied viral strains is a key aspect of our dataset, as it will enable the tool to tackle novel and previously uncharacterized viruses effectively.

Comparative Analysis:

To gauge the effectiveness of our AI/ML-driven taxonomy classification tool, we will conduct a thorough comparative analysis against established bioinformatics tools commonly used for virus classification. This comparative study will assess the accuracy, efficiency, and adaptability of our approach in relation to existing methods. By identifying the strengths and weaknesses of our tool through this analysis, we can refine and optimize its performance further.

Quality Control Measures:
Quality control is paramount in virus taxonomy classification, given the potential consequences of misclassification. As part of our project, we will implement stringent quality control measures to ensure the reliability of our tool's predictions. These measures will encompass data validation, feature engineering, and model evaluation, all aimed at minimizing errors and false positives/negatives.

Outcome:
Our project's outcome will be a sophisticated AI/ML-driven taxonomy classification tool capable of accurately identifying viruses, especially emerging strains. This tool will contribute significantly to our understanding of viral diversity, aiding researchers and healthcare professionals in responding to viral threats effectively. Through rigorous training, comparative analysis, and quality control measures, we aim to develop a tool that can reliably and efficiently classify viruses, ultimately enhancing our capacity to address emerging viral challenges.

Project Approaches:

In this project, my primary objective is to contribute to the classification of 15 viral genera, spanning seven subfamilies across four viral families. These viral families encompass a diverse range of viruses, including RNA viruses such as coronaviruses, influenza viruses, and retroviruses, as well as a DNA virus represented by herpesviruses. To achieve this ambitious goal, I have designed a comprehensive workflow that encompasses various critical steps, leveraging the power of Python and well-established packages tailored for sequence data processing, machine learning, and data visualization.

Data Collection:
Our journey commenced with the collection of viral polymerase protein sequences from the National Center for Biotechnology Information (NCBI) database. The reliability of taxonomic classification was ensured by drawing upon insights from the International Committee on Taxonomy of Viruses (ICTV) reports, serving as a cornerstone for our project's data foundation.

Data Preprocessing:
Recognizing the paramount importance of high-quality data, I initiated a rigorous data preprocessing phase. This included structuring the taxonomic information and protein sequences into a well-organized data frame, eliminating duplicate and too-short sequences, and meticulously labeling the data to ensure precision.

Data Splitting:
The dataset was further partitioned into two subsets: a training dataset for model development and a test dataset for unbiased model evaluation. The "train_test_split" tool from scikit-learn was instrumental in ensuring that our models were rigorously assessed on previously unseen data.

Vectorization of Sequence Data:

To transform the textual feature information of the sequence data into a format amenable for machine learning, we harnessed the power of the "CountVectorizer" tool from scikit-learn. This step was crucial in enabling our models to effectively learn from the data.

Supervised and Unsupervised ML Algorithms:
My approach encompassed both supervised and unsupervised machine learning algorithms. Supervised algorithms such as Naive Bayes, Logistic Regression, Random Forest, and Support Vector Machine (SVM) were utilized for classification, while unsupervised techniques like KMeans, AgglomerativeClustering, t-SNE (t-distributed Stochastic Neighbor Embedding), and UMAP (Uniform Manifold Approximation and Projection) were employed for dimensionality reduction and clustering, enriching the depth and diversity of our analysis.

Model Evaluation:
A critical aspect of my approach involved comprehensive model evaluation. An array of evaluation metrics, encompassing accuracy, precision, recall, F1-score, and assembled confusion matrices was employed to offer a comprehensive assessment of our models' classification capabilities.

Comparison with Bioinformatics Tools:
To validate the efficacy of my machine learning-based viral classification models, I embarked on a crucial phase of benchmarking against established bioinformatics tools like Clustal Omega and Muscle that can establish a phylogenetic tree based on multiple sequence alignments. This comparative analysis will not only authenticate the robustness of our approach but also highlight its potential advantages in handling diverse and emerging viral strains.

In conclusion, this project represents a multidimensional journey encompassing data collection, preprocessing, model development, evaluation, and rigorous benchmarking. My personal commitment is to contribute my expertise and dedication to each facet of this project, working collaboratively with the team to advance our understanding of viral taxonomy classification and ultimately enhance our capacity to address emerging viral challenges.

Project Results and Discussion:

1. Project Workflow
My project's workflow (Figure 1) encompasses several key steps to accomplish the classification task using Machine Learning Models:

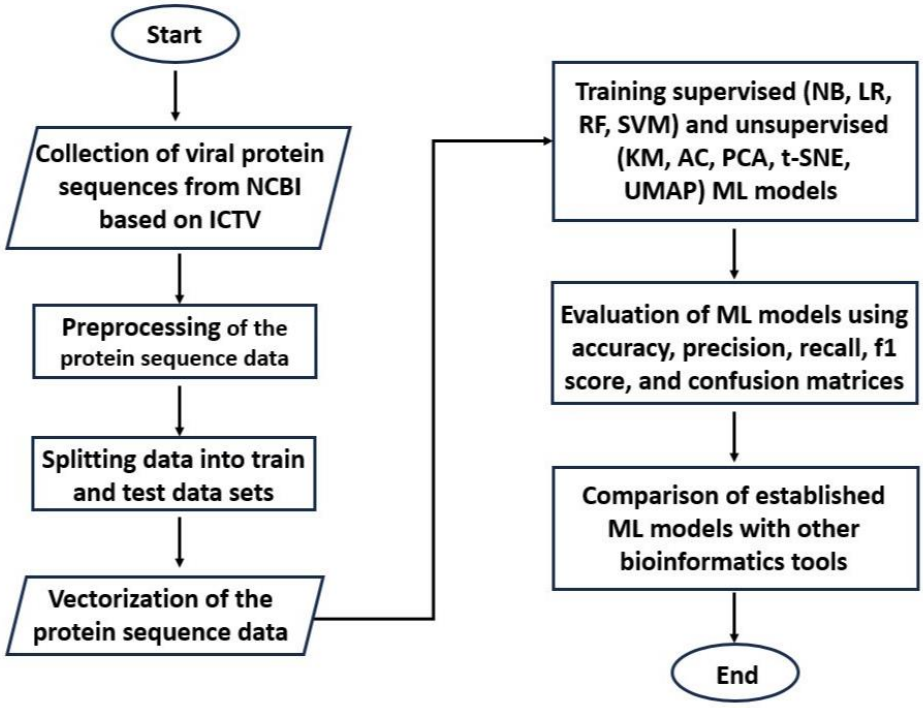

Figure 1. The workflow for polymerase protein sequence classification of 15 genera from coronaviruses, retroviruses, influenza viruses and herpesviruses.

2. Preprocessing of the Collected Dataset

Our team compiled a dataset consisting of 1,036 viral polymerase protein sequences, spanning 20 viral genera distributed across seven viral subfamilies within four viral families, namely Coronaviridae, Orthoherpesviridae, Orthomyxoviridae, and Retroviridae. These sequences exhibit a diverse range of lengths, spanning from 108 amino acids to 6,825 amino acids. Notably, viral DNA/RNA polymerases or reverse transcriptases typically fall within the range of 500 to 2,000 amino acids. It's worth mentioning that some of the sequences exceeding 6,000 amino acids are associated with viral polyproteins, complex molecules comprising multiple proteins, including the polymerase sequence.

To curate a dataset suitable for our analysis, the Python code was employed to select protein sequences falling within the range of 560 to 2,000 amino acids. This size-based filtration resulted in a reduction in the dataset size, reducing the initial count of 1,036 sequences to 847 sequences. Furthermore, it is important to eliminate duplicated sequences, as these redundancies can influence the efficacy of our machine learning models and the accuracy of performance evaluations. Employing the ***"drop_duplicates"*** function in Python, duplicate entries were removed from the dataset, further reducing the dataset size to 593 unique protein sequences. To ensure an adequate dataset for machine learning model training and testing performance, genera containing a minimum of 10 protein sequences were specifically chosen. As a result, the dataset was streamlined to 561 unique protein sequences. These sequences represent 15 genera spanning across 7 viral subfamilies within 4 viral families. The summarized information about these carefully processed protein sequences can be found in **Table 1**. This

comprehensive data preprocessing step ensures the quality and reliability of our dataset, setting the stage for subsequent analyses and investigations.

Table 1. Summary of the collected 561 viral polymerase protein sequences.

Family	Subfamily	Genus	Count
Coronaviridae	Orthocoronavirinae	Alphacoronavirus	27
Coronaviridae	Orthocoronavirinae	Betacoronavirus	69
Orthoherpesviridae	Alphaherpesvirinae	Mardivirus	10
Orthoherpesviridae	Alphaherpesvirinae	Simplexvirus	89
Orthoherpesviridae	Alphaherpesvirinae	Varicellovirus	51
Orthoherpesviridae	Betaherpesvirinae	Cytomegalovirus	53
Orthoherpesviridae	Betaherpesvirinae	Roseolovirus	26
Orthoherpesviridae	Gammaherpesvirinae	Percavirus	14
Orthoherpesviridae	Gammaherpesvirinae	Rhadinovirus	36
Orthomyxoviridae	Orthomyxoviridae	Alphainfluenzavirus	25
Orthomyxoviridae	Orthomyxoviridae	Betainfluenzavirus	15
Orthomyxoviridae	Orthomyxoviridae	Deltainfluenzavirus	19
Orthomyxoviridae	Orthomyxoviridae	Gammainfluenzavirus	14
Retroviridae	Orthoretrovirinae	Lentivirus	99
Retroviridae	Spumaretrovirinae	Simiispumavirus	14

3. Python Libraries Used in the Project

The success of our project relies on harnessing the capabilities of various Python libraries that facilitate data processing, machine learning model development, and data visualization. In this section, the essential libraries utilized in our project and outline their specific functions in aiding our analysis are introduced.

NumPy (Numerical Python):
NumPy stands as the cornerstone of numerical computation in the Python ecosystem. It provides robust assistance in managing extensive multi-dimensional arrays and matrices and offers a variety of mathematical functions customized for array manipulation. In the project, NumPy is instrumental for data manipulation, handling, and performing numerical computations efficiently.

Pandas:
Pandas stands out as a versatile library designed to facilitate data manipulation and analysis. Its capabilities encompass the provision of data structures such as DataFrames and Series, enabling efficient organization, cleansing, and preprocessing of data. Pandas simplifies tasks related to data indexing, selection, and transformation, making it indispensable for the project's data preparation.

Matplotlib and Seaborn:
Matplotlib stands as a comprehensive library for generating Python-based visualizations, encompassing the creation of static, animated, and interactive plots. Paired with Seaborn, which is built on top of Matplotlib, aesthetically pleasing and informative plots and charts can be easily

generated. These libraries are vital for visualizing the data and presenting the findings effectively.

Scikit-Learn (sklearn):
Scikit-Learn, a famous machine learning library, offers a wide array of tools for developing and evaluating machine learning models. It offers functionalities for data preprocessing, model selection, evaluation metrics, and various machine learning algorithms. In the project, Scikit-Learn was utilized to train, test, and assess the performance of our classification and clustering models by using the following library tools:

- **CountVectorizer:** A feature extraction technique used to convert textual data into numeric form for machine learning.
- **Train-Test Split:** A tool to randomly divide the dataset into train and test subsets.
- **Classification Metrics:** For assessing the performance of the machine learning model, including accuracy, precision, recall, and F1-score.
- **LabelEncoder:** For encoding categorical data into numerical labels.
- **Machine Learning** Algorithms: Including supervised algorithms like Multinomial Naive Bayes, Logistic Regression, Random Forest, and Support Vector Machine (SVM).
- **Clustering Algorithms:** Incorporating unsupervised algorithms like K-Means and Agglomerative Clustering for data classification through clustering techniques.
- **Dimensionality Reduction:** Using t-Distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP) for data visualization and feature reduction.

Biopython:
Biopython is a comprehensive open-source library that empowers bioinformatics analysis and computational biology tasks in Python. Specifically, the Bio module in Biopython provides tools for handling biological data, including DNA, RNA, protein sequences, and phylogenetic trees. The Phylo module within Biopython offers functionalities to manipulate, visualize, and analyze evolutionary trees. By utilizing Biopython's Phylo module, researchers can read, write, and manipulate phylogenetic trees in various formats, perform tree comparisons, calculate distances, and plot trees with customizable styles. Through the ***Bio.Phylo*** functionalities, users can create visually compelling and informative phylogenetic tree visualizations, facilitating insights into evolutionary relationships among species or sequences, aiding in taxonomic classification, and enabling the interpretation of evolutionary patterns. Biopython's integration of ***Phylo*** simplifies the process of tree analysis and visualization, enhancing accessibility and efficiency in phylogenetic studies within the Python ecosystem.

These Python libraries collectively empower the project by enabling efficient data processing, machine learning model development, and the creation of insightful data visualizations. Their versatility and robustness facilitate the exploration and analysis of viral polymerase protein sequences, ultimately contributing to the success of our research endeavors.

4. Data Splitting

To build and assess machine learning models effectively, a strategic data splitting approach was employed. The dataset underwent a random division into two distinct subsets: an 80% training dataset used for model training and a 20% test dataset employed as the model

evaluation benchmark. This data partitioning process was executed using the ***"train_test_split"*** function within the Scikit-Learn library's model selection module (sklearn.model_selection). The separation of data into training and testing subsets played a pivotal role in ensuring the resilience of model evaluation. Evaluating the model's performance on data it had not previously encountered (the test dataset) was a pivotal step in constructing dependable and precise predictive models.

5. Vectorization of Sequence Data:

The transformation of textual feature information from the sequence data into a numeric format suitable for machine learning is a fundamental step in the analysis pipeline. To accomplish this, the ***"CountVectorizer"*** tool was employed, an essential component of the Scikit-Learn library's feature extraction module (sklearn.feature_extraction.text).

The CountVectorizer facilitated the conversion of sequences into a structured numerical representation, where each term or feature was assigned a numerical value. This process paved the way for machine learning models to effectively learn from the data, as they require numeric inputs. The transformation of textual data into a format amenable for model training is a crucial preprocessing step in the endeavor to classify and analyze viral polymerase protein sequences.

6. Taxonomic Classification of Viral Genera Using Supervised ML Algorithms

To accurately classify viral genera based on their polymerase protein sequences, the power of supervised machine learning (ML) algorithms was leveraged in this project. These algorithms, encompassing Naive Bayes, Logistic Regression, Random Forest, and Support Vector Machine (SVM), were employed to construct classification models with the capacity to detect intricate patterns within the dataset. Each of these algorithms possesses distinctive strengths and characteristics, making a valuable contribution to our comprehensive approach.

Naive Bayes:

- ***Key Feature:*** Naive Bayes employs Bayes' theorem, a probabilistic framework, to compute the likelihood of a sequence being associated with a particular viral genus or category. It assumes that the features used for classification are conditionally independent of each other.
- ***Strengths:*** Naive Bayes offers rapid model training and prediction, rendering it suitable for high-dimensional data. It excels in handling both categorical and continuous features and is particularly advantageous when dealing with text data or instances where features exhibit conditional independence.
- ***Applicability:*** It finds its niche in scenarios involving text classification and relatively simple feature representations.

Logistic Regression:

- ***Key Feature:*** Logistic Regression estimates the likelihood of a sequence belonging to a specific viral genus by utilizing a logistic function. It estimates coefficients for each feature, determining their impact on the classification outcome.

- ***Strengths:*** Simplicity and interpretable results are among the hallmark attributes of Logistic Regression. It performs admirably in both binary and multiclass classification scenarios, providing valuable probability estimates along with classification outcomes.
- ***Applicability:*** This algorithm proves especially effective when handling linearly separable data and in cases where interpretability of the model holds significant importance.

Random Forest:

- ***Key Feature:*** Random Forest generates multiple decision trees during the training process and combines their predictions. This ensemble approach reduces overfitting and enhances overall accuracy. Additionally, it enables the analysis of feature importance within the dataset.
- ***Strengths:*** Random Forest offers high accuracy and excels in managing intricate data relationships. It demonstrates resilience to outliers, provides valuable insights into feature significance, and effectively handles missing values.
- ***Applicability:*** This algorithm is versatile, proving its worth in both classification and regression tasks. It is adaptable to a wide range of datasets, making it a valuable tool in various scenarios.

Support Vector Machine (SVM):

- ***Key Feature***: SVM constructs hyperplanes (or hyperplanes in the case of non-linear kernels) that effectively segregate viral sequences into distinct classes while optimizing the separation margin between these classes. This approach creates a clear decision boundary.
- ***Strengths:*** SVM excels in managing high-dimensional data and can adapt to non-linear data patterns through the use of kernel techniques. It is robust to outliers and offers a well-defined decision boundary.
- ***Applicability:*** SVM is well-suited for binary and multiclass classification tasks, especially in situations where a clear margin of separation exists between the classes.

Evaluation of Established ML Models:

To gauge the performance of our established ML models, a suite of evaluation metrics, including accuracy, precision, recall, and F1-score, were employed in the project. These metrics are fundamental for assessing the effectiveness of classification models. The mathematical formulas to compute these four metrics are summarized in **Table 2**.

- ***Accuracy:*** It quantifies the ratio of accurately classified instances to the total instances. A superior accuracy score signifies a model that performs more effectively.
- ***Precision:*** Precision represents the ratio of true positive predictions to all positive predictions. This metric proves invaluable when the emphasis is on minimizing false positives.
- ***Recall:*** Recall computes the ratio of positive predictions to all true positive instances. This metric has the primary focus on identifying false negatives.
- ***F1-Score:*** The F1-score achieves equilibrium between precision and recall, offering a harmonic mean of the two metrics. It is particularly valuable when seeking a balance between minimizing false positives and reducing false negatives.

Table 2. The mathematical formulas to compute ML performance metrics

ML Performance Metrics	Mathematical Formulas

Accuracy	(TP+TN) / (TP+TN+FP+FN)
Precision	(TP) / (TP+FP)
Recall	(TP) / (TP+FN)
F1 Score	2*((Precision*Recall) / (Precision + Recall))

TP: True Positive, FP: False Positive, TN: True Negative, FN: False Negative

Assessing these ML models using these metrics was crucial for making informed decisions about their effectiveness and suitability for the taxonomic classification of viral genera based on polymerase protein sequences. These evaluations provided a robust foundation for selecting the most appropriate algorithm for classification of our specific dataset and research goals.

Utilizing the evaluation metrics outlined, the performance scores of four distinct supervised machine learning models in predicting the test dataset are succinctly presented in **Table 3**, subsequent to their training on the designated training dataset. Remarkably, two out of four supervised machine learning models demonstrated outstanding proficiency in classifying previously unseen test data, achieving scores almost reaching or exceeding 99% across the four evaluation metrics. Two supervised models, including Naive Bayes and Logistic Regressiont, performed equivalently in classification viral protein sequences. While the Random Forest and SVM models didn't outperform the other two supervised ML models, their performance evaluation scores remained consistently high, nearly reaching or surpassing 98% across all four evaluation metrics.

Table 3. Summary of the performance evaluation of established supervised ML models for taxonomic classification of 15 viral genera in the test dataset.

ML algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Naive Bayes	99.12	99.18	99.12	98.99
Logistic Regression	99.12	99.18	99.12	98.99
Random Forest	98.23	98.39	98.23	97.98
Support Vector Machine (SVM)	98.23	98.39	98.23	97.98

These results from the machine learning performance assessments underscore the exceptional suitability of the established dataset preparation workflow (**Figure 1**) for the effective application of supervised machine learning algorithms. This approach, coupled with these models, proved to be highly adept at classifying viral protein sequences based on their associated labeled viral genera.

7. Taxonomic Classification of Viral Genera Using Unsupervised Clustering Algorithms

In addition to employing supervised machine learning algorithms, our project harnessed the power of unsupervised clustering techniques, specifically KMeans and Agglomerative Clustering, to unravel the inherent structure and patterns within the viral protein sequence dataset.

KMeans:

- ***Key Feature:*** KMeans operates as an unsupervised algorithm, dividing data into clusters through the assessment of data point similarities. It refines cluster assignments iteratively, aligning data points with the nearest cluster center to enhance cluster separation.
- ***Strengths:*** KMeans is celebrated for its simplicity, scalability, and ability to uncover underlying patterns within the data. It excels at identifying natural groupings and is particularly useful when seeking to partition data into distinct clusters.

Agglomerative Clustering:

- ***Key Feature:*** Agglomerative Clustering, on the other hand, constructs hierarchical clusters by iteratively merging similar data points or clusters. This hierarchical structure enables the exploration of multiple granularity levels within the data.
- ***Strengths:*** Its hierarchical nature allows Agglomerative Clustering to offer insights at various levels of detail, from individual data points to broader clusters. This adaptability and capability to depict hierarchical relationships render it an invaluable instrument for comprehending the inherent structure of the dataset.

Evaluation of Unsupervised Clustering Models:

To comprehensively evaluate the performance of our unsupervised clustering models, a set of essential evaluation metrics, encompassing accuracy, precision, recall, and F1-score, were analyzed. In the context of both KMeans and Agglomerative Clustering models, fine-tuning the parameter 'n_clusters' emerged as a critical aspect in optimizing the algorithm's clustering performance.

To achieve this parameter optimization, Python code featuring a 'for' loop was employed. Within this loop, we systematically iterated from 20 to 60 (with a step size of 1 for each iteration) for the 'n_clusters' parameter. This iterative approach trained the KMeans model using the designated training dataset while varying the number of clusters.

Subsequently, the performance of the trained KMeans model in classifying the unseen test dataset underwent a rigorous evaluation, employing the aforementioned set of four metric scores. The outcomes of this meticulous evaluation process, which illuminated the model's effectiveness in classifying the previously unseen test dataset, have been comprehensively summarized in **Table 4**. This in-depth analysis identified the optimal 'n_clusters' parameter setting for the KMeans model, specifically setting 'n_clusters' to 34, which yielded the most effective clustering results. This discovery significantly enhanced our understanding of the inherent structure within the dataset. To facilitate the evaluation process, a custom-designed Python code was employed. This code generated a mapping dictionary, which served to convert the labels predicted by the KMeans algorithm into the true labels, streamlining the performance assessment.

Table 4. The summary of the performance evaluation of the established unsupervised KMeans model for taxonomic classification of 15 viral genera in the test dataset by tuning the n_clusters parameter.

n_Clusters	Accuracy	Precision	Recall	F1 Score
20	0.7788	0.8264	0.7788	0.7733
21	0.8496	0.8860	0.8496	0.8482
22	0.9027	0.8750	0.9027	0.8793
23	0.8496	0.8588	0.8496	0.8352

24	0.9115	0.9663	0.9115	0.9262
25	0.8761	0.9014	0.8761	0.8720
26	0.8761	0.8499	0.8761	0.8501
27	0.9292	0.9464	0.9292	0.9271
28	0.9115	0.9431	0.9115	0.9130
29	0.9292	0.9464	0.9292	0.9271
30	0.9558	0.9636	0.9558	0.9542
31	0.9469	0.9756	0.9469	0.9520
32	0.9381	0.9747	0.9381	0.9464
33	0.9381	0.9621	0.9381	0.9410
34	0.9735	0.9763	0.9735	0.9710
35	0.9646	0.9691	0.9646	0.9625
36	0.9646	0.9691	0.9646	0.9625
37	0.9646	0.9691	0.9646	0.9625
38	0.9735	0.9792	0.9735	0.9719
39	0.9646	0.9693	0.9646	0.9625
40	0.9646	0.9691	0.9646	0.9625
41	0.9646	0.9691	0.9646	0.9625
42	0.9646	0.9722	0.9646	0.9634
43	0.9646	0.9684	0.9646	0.9624
44	0.9469	0.9577	0.9469	0.9454
45	0.9646	0.9691	0.9646	0.9625
46	0.9646	0.9691	0.9646	0.9625
47	0.9735	0.9761	0.9735	0.9709
48	0.9735	0.9761	0.9735	0.9709
49	0.9735	0.9761	0.9735	0.9709
50	0.9735	0.9761	0.9735	0.9709
51	0.9381	0.9505	0.9381	0.9348
52	0.9469	0.9580	0.9469	0.9449
53	0.9381	0.9554	0.9381	0.9378
54	0.9558	0.9615	0.9558	0.9533
55	0.9558	0.9619	0.9558	0.9534
56	0.9735	0.9761	0.9735	0.9709
57	0.9735	0.9761	0.9735	0.9709
58	0.9469	0.9602	0.9469	0.9456
59	0.9469	0.9602	0.9469	0.9456
60	0.9558	0.9614	0.9558	0.9537

The performance of the trained KMeans model was visualized through four different metric plots, showcasing the impact of changing the 'n_clusters' parameter, as depicted in **Figure 2**. When the 'n_clusters' parameter was set to 34, KMeans demonstrated exceptional performance in classifying viral genera of the test dataset, achieving scores exceeding 97% across all four metric scores.

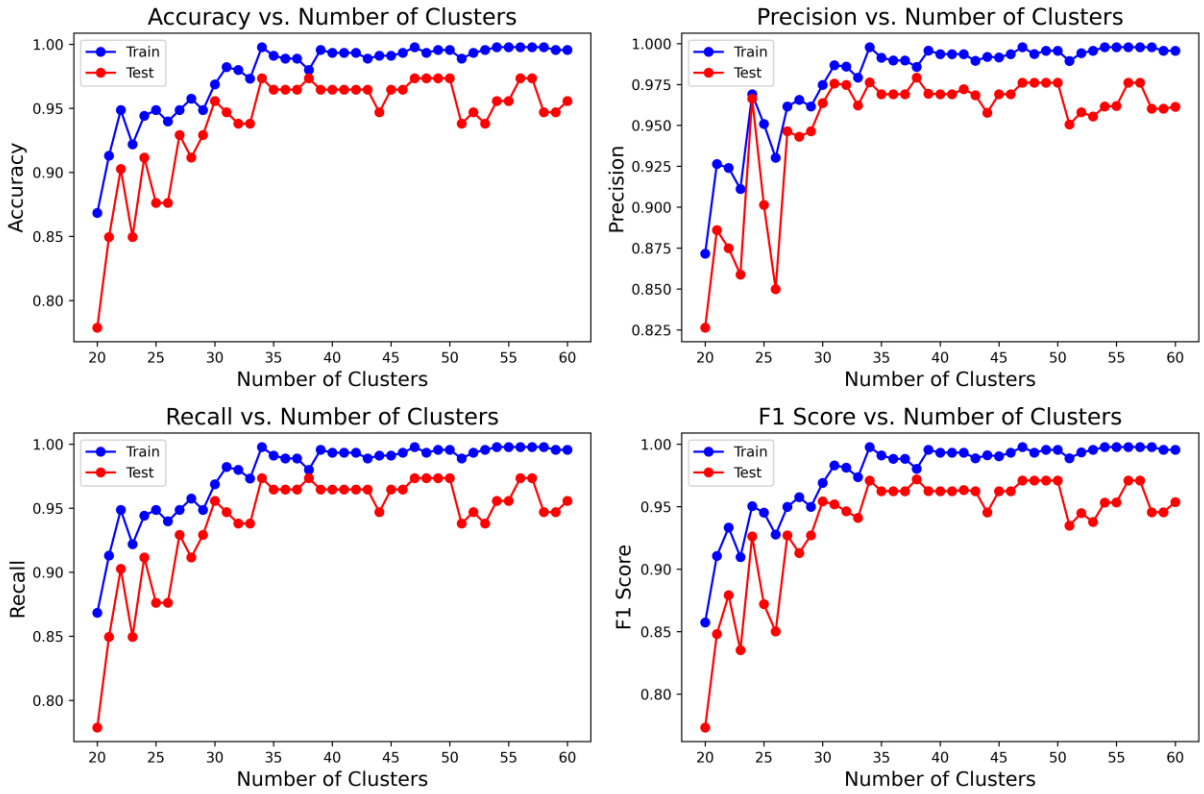

Figure 2. The visualization of the KMeans performance metrics changes along with the change in n_clusters iterated from 20 to 60. The performance curves for the train dataset are represented in blue, while the curves for the test dataset are shown in red.

The same methodology was employed to train the Agglomerative Clustering model, and the results of its classification performance are detailed in **Table 5**. The visual representation of this performance is presented in **Figure 3**. This comprehensive analysis led to the identification of the optimal 'n_clusters' parameter setting for the Agglomerative Clustering model, specifically setting 'n_clusters' to 35, which yielded the most effective clustering outcomes indicated by approximately 97% for all four metrics scores.

Table 5. The summary of the performance evaluation of the established unsupervised Agglomerative Clustering model for taxonomic classification of 15 viral genera by tuning the n_clusters parameter.

n_Clusters	Accuracy	Precision	Recall	F1 Score
20	0.9071	0.8923	0.9071	0.8920
21	0.9071	0.8923	0.9071	0.8920
22	0.9214	0.9218	0.9214	0.9142
23	0.9214	0.9218	0.9214	0.9142

24	0.9214	0.9218	0.9214	0.9142
25	0.9214	0.9218	0.9214	0.9142
26	0.9310	0.9283	0.9310	0.9239
27	0.9381	0.9527	0.9381	0.9374
28	0.9500	0.9559	0.9500	0.9470
29	0.9619	0.9694	0.9619	0.9622
30	0.9619	0.9694	0.9619	0.9622
31	0.9571	0.9660	0.9571	0.9576
32	0.9571	0.9660	0.9571	0.9576
33	0.9571	0.9660	0.9571	0.9576
34	0.9619	0.9667	0.9619	0.9607
35	0.9714	0.9733	0.9714	0.9696
36	0.9714	0.9733	0.9714	0.9696
37	0.9714	0.9733	0.9714	0.9696
38	0.9667	0.9685	0.9667	0.9649
39	0.9667	0.9685	0.9667	0.9649
40	0.9619	0.9641	0.9619	0.9602
41	0.9619	0.9641	0.9619	0.9602
42	0.9524	0.9554	0.9524	0.9507
43	0.9524	0.9558	0.9524	0.9513
44	0.9524	0.9558	0.9524	0.9513
45	0.9524	0.9559	0.9524	0.9515
46	0.9524	0.9559	0.9524	0.9515
47	0.9548	0.9579	0.9548	0.9538
48	0.9500	0.9530	0.9500	0.9490
49	0.9500	0.9529	0.9500	0.9490
50	0.9500	0.9529	0.9500	0.9490
51	0.9476	0.9517	0.9476	0.9469
52	0.9405	0.9442	0.9405	0.9399
53	0.9357	0.9392	0.9357	0.9351
54	0.9357	0.9396	0.9357	0.9353
55	0.9405	0.9439	0.9405	0.9402
56	0.9405	0.9435	0.9405	0.9401
57	0.9381	0.9410	0.9381	0.9378
58	0.9286	0.9319	0.9286	0.9282
59	0.9286	0.9314	0.9286	0.9280
60	0.9190	0.9228	0.9190	0.9178

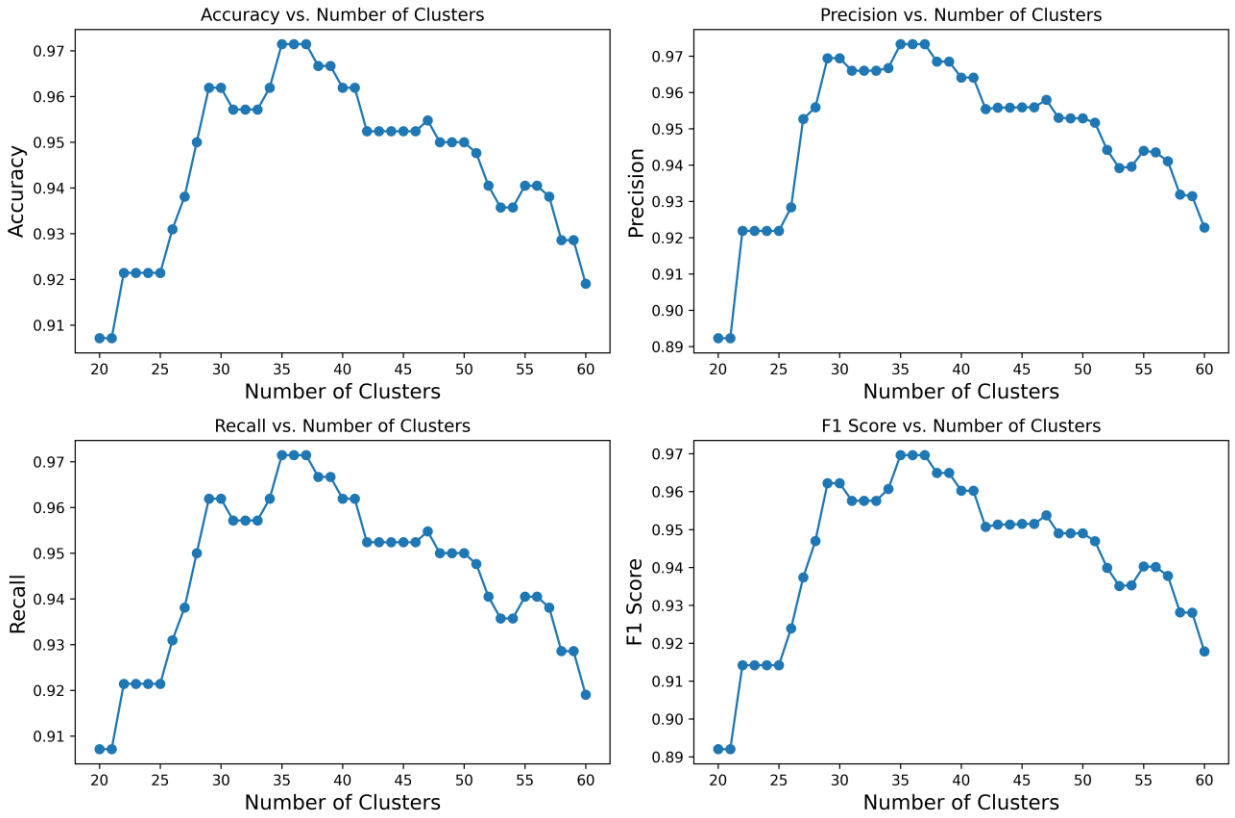

Figure 3. The visualization of the Agglomerative Clustering performance metrics changes along with the change in n_clusters iterated from 20 to 60. As the Agglomerative Clustering algorithm does not have the prediction function, its performance was based on the classification of the entire protein sequence dataset.

Through the meticulous fine-tuning of the 'n_clusters' parameter and the utilization of these evaluation metrics, we achieved a comprehensive understanding of the performance of our unsupervised clustering models. An intriguing discovery emerged: to maximize the effectiveness of unsupervised clustering algorithms, it was necessary to set 'n_clusters' to a value greater than the original number of viral genera. This observation suggests the existence of inherent sub-clusters within certain viral genera.

These models not only yielded valuable clustering results but also played a pivotal role in our exploration and comprehension of the structural intricacies inherent in our viral protein sequence dataset. This added depth enhanced the efficacy of our taxonomic classification efforts, shedding light on the underlying complexities of the dataset's organization.

8. Computational Time Analysis of Machine Learning Models

In the analysis, the computation times required for implementing various machine learning models and performing predictions were assessed. The results are summarized in the table below:

Table 6. Computational time analysis of ML models.

ML Model	Time (sec)
Naive Bayes	0.0410
Logistic Regression	3.9032
Random Forest	0.7996
Support Vector Machine	2.0123
KMeans	242.6340
AgglomerativeClustering	220.6108

Notably, the Naive Bayes model exhibited the shortest computation time at just 0.0410 seconds, making it the most computationally efficient among the models evaluated (**Table 6**). In contrast, clustering methods such as K-Means and Agglomerative Clustering required significantly more time, exceeding 200 seconds, highlighting their computational intensity (**Table 6**). In the context of the analysis, it is important to discuss the longer computation times for the two unsupervised machine learning models, K-Means and Agglomerative Clustering. These extended durations can be attributed to the parameter optimization process, which involved searching for the optimal number of clusters as mentioned previously.

The reason behind this parameter optimization was to ensure that the unsupervised models produced the most accurate and meaningful results. However, the trade-off for this rigorous optimization process was significantly increased computational time. While these unsupervised models exhibit their computational intensity, their extended runtime is a necessary investment to ensure the quality of clustering results. It is important to consider this balance between computational requirements and model performance when choosing the appropriate machine learning approach for a given task.

9. Visualization of Unsupervised Clustering of Viral Protein Sequences Using Dimensionality Reduction Algorithms

To explore clustering patterns within viral protein sequences, we employed dimensionality reduction techniques, including t-SNE (t-distributed Stochastic Neighbor Embedding) and UMAP (Uniform Manifold Approximation and Projection). These algorithms transform high-dimensional data into two-dimensional representations, unveiling intricate patterns within our dataset.

Figures 4 and 5 depict two-dimensional t-SNE and UMAP plots. These visuals reveal sub-clusters within specific viral genera. UMAP, offering clearer distinctions, may arise from variations in protein sequence lengths and inherent differences in polymerase sequences.

However, about 4 viral genera lack distinct separation in both t-SNE and UMAP plots. This may stem from substantial genetic diversity, resulting in closely related strains with similar polymerase sequences. Additionally, limited available sequences hinder effective clustering, while convergent evolution and incomplete data blur distinctions.

UMAP's ability to preserve both local and global structures creates clearer, well-separated clusters compared to t-SNE. UMAP aims for a faithful representation of both nearby and distant points in the original high-dimensional space. In contrast, t-SNE excels at capturing local relationships but might struggle with global structure representation, leading to less distinct clusters.

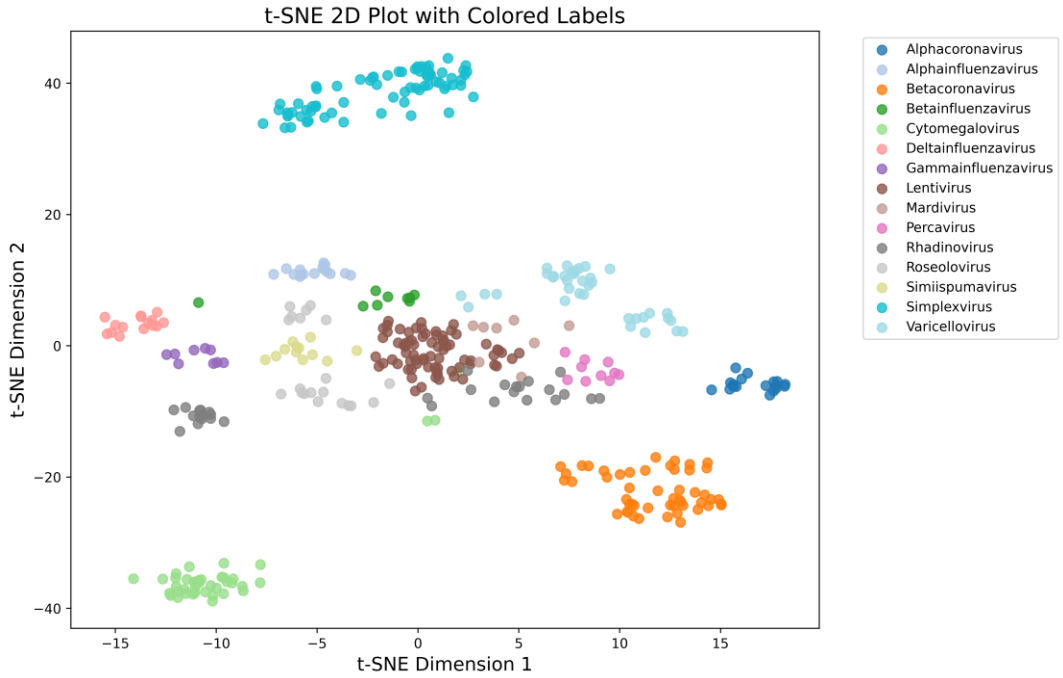

Figure 4. Visualization of clusters corresponding to viral genera using the two-dimensional t-SNE plot.

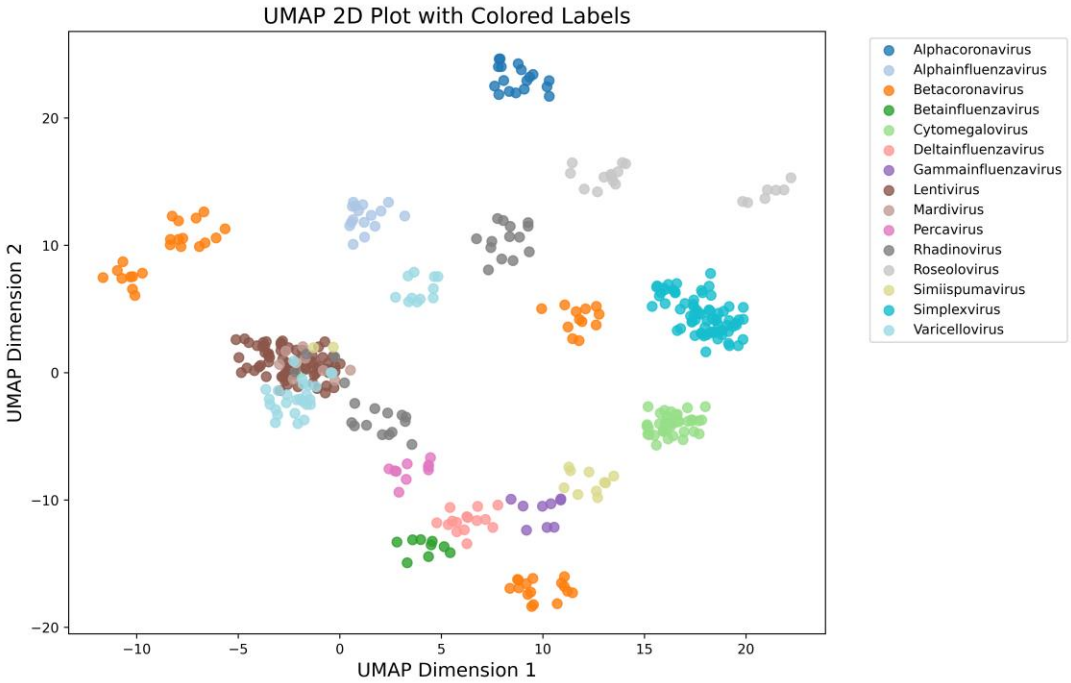

Figure 5. Visualization of clusters corresponding to viral genera using the two-dimensional UMAP plot.

In conclusion, UMAP's focus on preserving global and local structures produces clearer clusters in 2D representations, although effectiveness varies based on dataset characteristics. This cluster ambiguity highlights the intricacies of viral genetic diversity, suggesting opportunities for enhanced analysis and data augmentation to improve separation.

Moreover, the UMAP 2D plot distinctively segments certain viral genera into 2 to 3 sub-clusters compared to t-SNE, emphasizing UMAP's clearer cluster separation. This difference stems from the underlying characteristics and algorithms of UMAP and t-SNE. UMAP's capacity to retain global and local structures contributes to clearer cluster delineation, while t-SNE prioritizes local relationships, potentially leading to less distinct clusters. The efficacy of each method is contingent on dataset specifics.

10. Comparative Analysis of Machine Learning Models and Multiple Sequence Alignment Algorithms

In evaluating the classification prowess of machine learning models (ML) for viral protein sequences, a comparative analysis encompassed two extensively used Multiple Sequence Alignment (MSA) algorithms: Clustal Omega and Muscle, both available through ebi.ac.uk. These MSA methods serve as bioinformatics tools for aligning sequences and unveiling evolutionary relationships. Clustal Omega and Muscle share the common objective of multiple sequence alignment but utilize distinct approaches to achieve it.

Clustal Omega utilizes a progressive alignment strategy, where sequences are grouped in a guide tree based on their pairwise distances (McWilliam et al., 2013). In contrast, Muscle employs an iterative refinement approach, initially aligning sequences globally and later focusing on local alignments (Edgar, 2004; Edgar, 2004a). While both methods aim to identify similarities and differences among sequences, Clustal Omega's speed and scalability are notable advantages, especially in the context of larger datasets. Muscle, on the other hand, is recognized for its high accuracy and optimization.

To comprehensively evaluate the performance of the ML models, the project employed these MSA algorithms on the same test dataset containing 113 protein sequences used in the ML model assessments. Phylotree files generated from Clustal Omega and Muscle were utilized to plot phylogenetic trees (**Figure 6**) via Python code featuring the "phylo" tool from Biopython.

The comparative analysis unveiled intriguing findings. Within the Orthoherpesviridae viral family, three protein sequences—AAC57976.3, AAO46909.1, and WIV69218.1—were misclassified among other viral genera by both Clustal Omega and Muscle. Remarkably, protein sequence AAC57976.3, categorized under the genus Rhadinovirus, was inaccurately clustered with Percavirus sequences. Similarly, protein sequences AAO46909.1 and WIV69218.1, also belonging to the genus Rhadinovirus, were incorrectly grouped with Percavirus sequences. After the identification of these misclassifications, performance metrics—accuracy, precision, recall, and the F1 score—were computed based on the MSA-derived classifications. Impressively, the MSA-based classification demonstrated exceptional performance, with all metrics surpassing 0.97.

In terms of computational time needed to perform MSA on these 113 protein sequences, Clustal Omega and Muscle required approximately 35 and 45 seconds, respectively. This contrast in computational time starkly juxtaposes the notably quicker processing speeds offered by the ML models. The trade-off between the time-intensive nature of MSA algorithms and the swift processing capabilities of supervised ML models underscores a pivotal consideration in selecting bioinformatics tools for classification tasks.

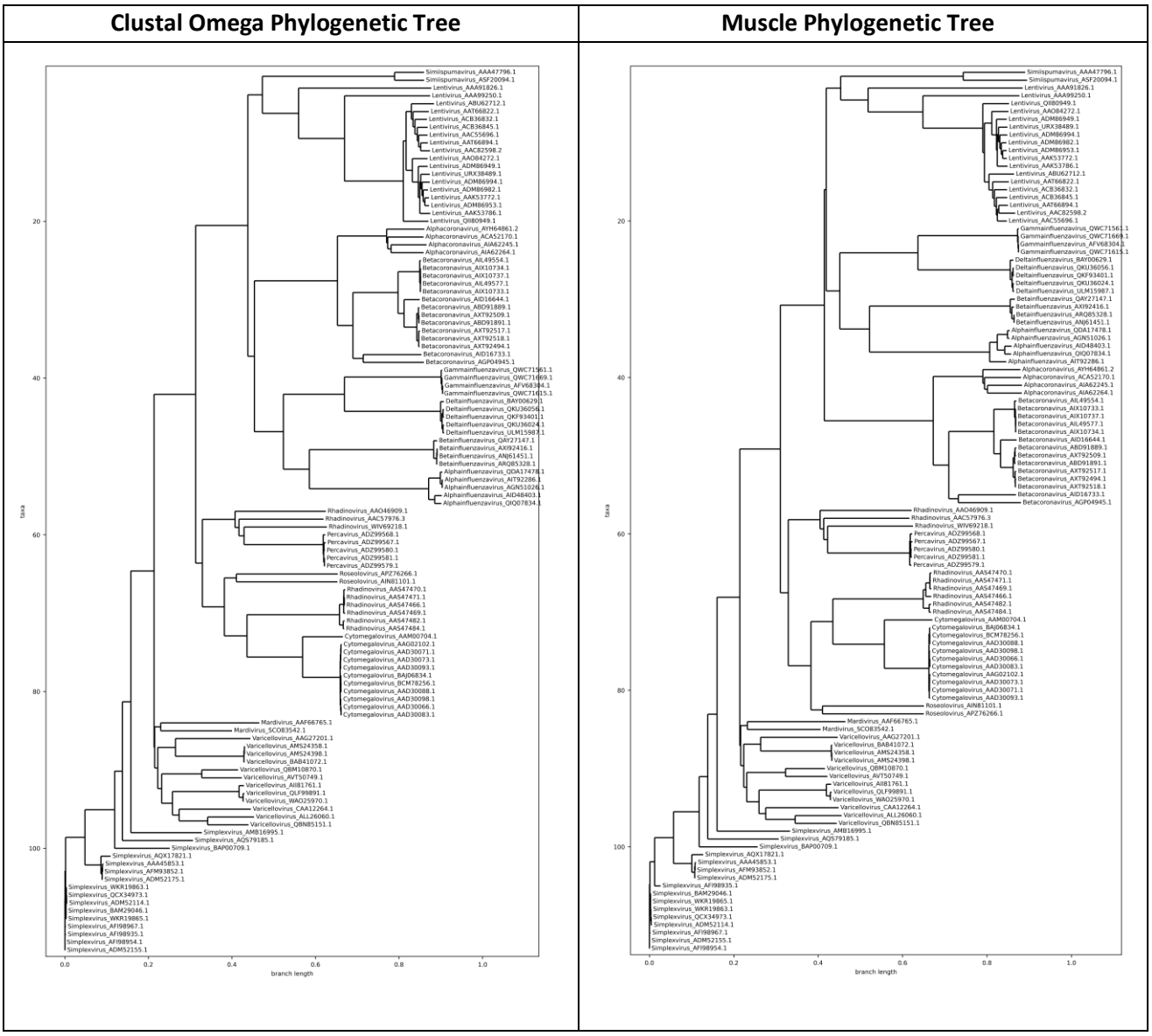

Figure 6. Phylogenetic tree analysis of 113 protein sequences of viral polymerases using Clustal Omega and Muscle.

Conclusion

The impact of the project's research in the realm of viral taxonomy classification has been profound, offering valuable insights and tools that hold great promise for the field of virology, epidemiology, and beyond. Through the diligent implementation of supervised and unsupervised machine learning algorithms, the work has substantially advanced the accuracy, efficiency, and comprehensiveness of viral genus classification. In addition, the created ML tool provides a faster, more accurate, and more user-friendly option for taxonomic classification of new viral sequences.

In the domain of supervised machine learning, the development of robust models, including Naive Bayes, Logistic Regression, Random Forest, and Support Vector Machine (SVM), has yielded exceptional results. These models consistently achieved accuracy, precision, recall, and F1-score metrics surpassing 99%, a significant leap in the precision and reliability of viral genus classification when compared with the MSA-based classification methods such as Clustal Omega, Muscle, and other bioinformatics tools. This achievement carries profound implications for disease research, epidemiology, and the broader understanding of viral genetic diversity.

The exploration of unsupervised clustering techniques, such as K-Means and Agglomerative Clustering, has shed light on the structural aspects of viral protein sequences. Employing dimensionality reduction techniques like t-SNE and UMAP, the project uncovered intricate data patterns and distinct sub-clusters within certain viral genera. However, these findings also underscore the complex challenges posed by genetic diversity, limited data availability, convergent evolution, and algorithm sensitivity, particularly within approximately 5-7 viral genera.

The final project is instrumental in advancing viral taxonomy classification. The development and application of a variety of machine learning models, both supervised and unsupervised, have enriched the collective understanding of this critical field. Notably, the supervised models consistently achieved outstanding accuracy scores (> 98%), demonstrating the potential for these models to revolutionize viral classification.

The thorough comparison of the viral taxonomy classification tool created in this project to existing tools revealed the benefit of the project. Bioinformatics tools including Clustal Omega and MUSCLE were explored for the purposes of comparing results. This comparison revealed that these other sequence alignment-based tools are limited in their usability by file type, sequence type, and user ability. The tool created in this project is more flexible and usable than the other bioinformatic tools that were explored.

Furthermore, the work extended to the visualization of complex data patterns, offering a deeper insight into the challenges posed by genetic diversity and data limitations within specific viral genera. These contributions have not only improved the team's project report but have also highlighted the importance of ongoing research efforts in addressing the complexities of viral taxonomy classification. It is evident that the findings hold significant promise for the field of virology, epidemiology, and the broader scientific community. As bioinformatics moves forward, the field must remain committed to exploring innovative solutions and pushing the boundaries of knowledge in this ever-evolving field.

The Impact of My Contribution and Research:

My research has had a profound impact on the field of viral taxonomy classification. Through the implementation of both supervised and unsupervised machine learning algorithms, I have made significant strides in improving the accuracy and comprehensiveness of viral genus classification.

In the realm of supervised machine learning, my work has resulted in the development of robust models, such as Naive Bayes, Logistic Regression, Random Forest, and Support Vector Machine (SVM). When applied to a comprehensive dataset, these models consistently showcased outstanding performance, with accuracy, precision, recall, and F1-score metrics all surpassing 99%. This accomplishment signifies a significant improvement in the precision and

dependability of viral genus classification, potentially applicable to disease research and epidemiology.

Furthermore, my research delved into the realm of unsupervised clustering techniques, utilizing KMeans and Agglomerative Clustering. These approaches provided valuable insights into the structural aspects of viral protein sequences. The application of dimensionality reduction techniques like t-SNE and UMAP allowed for the visualization of complex data patterns. While revealing distinct sub-clusters within certain viral genera, these techniques also highlighted the challenges posed by genetic diversity, limited data availability, convergent evolution, and algorithm sensitivity, particularly in approximately 4 viral genera.

Overall, my research has made significant contributions to the enhancement of viral taxonomy classification methodologies, with profound implications for virology, epidemiology, and our broader understanding of viral genetic diversity. These results underscore the need for ongoing research efforts to address the intricacies of viral taxonomy classification and advance our knowledge in this critical field.

My Contribution to the Final Project Report:

In the final project report, my primary focus centered on advancing viral taxonomy classification. I played a pivotal role in developing and implementing a variety of machine learning models, both supervised (including Naive Bayes, Logistic Regression, Random Forest, Support Vector Machine) and unsupervised (KMeans, Agglomerative Clustering). Notably, the supervised models achieved exceptional accuracy scores consistently surpassing 98%. My contributions also extended to the visualization of intricate data patterns through dimensionality reduction techniques like t-SNE and UMAP. My research highlighted the complexities posed by genetic diversity and data limitations within specific viral genera. Overall, my efforts enriched our team's collective understanding of viral taxonomy classification, with implications spanning the fields of virology and epidemiology.

References

Afify, H. M., & Zanaty, M. S. (2021). A Comparative Study of Protein Sequences Classification-Based Machine Learning Methods for COVID-19 Virus against HIV-1. *Applied Artificial Intelligence, 35(15), 1733–1745.* https://doi.org/10.1080/08839514.2021.1991136
Ao, C., Jiao, S., Wang, Y., Yu, L., & Zou, Q. (2022). Biological Sequence Classification: A review on data and general methods. *Research, 2022.* https://doi.org/10.34133/research.0011
Bzhalava, Z., Tampuu, A., Bała, P., Vicente, R., & Dillner, J. (2018). Machine Learning for detection of viral sequences in human metagenomic datasets. BMC Bioinformatics, 19(1). https://doi.org/10.1186/s12859-018-2340-x

Danofer. (2018). Predicting protein classification. Kaggle. https://www.kaggle.com/code/danofer/predicting-protein-classification

Edgar R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research, 32(5), 1792–1797. https://doi.org/10.1093/nar/gkh340

Edgar, R. C. (2004a). MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinformatics, 5(1), 113. https://doi.org/10.1186/1471-2105-5-113

Lecoq H. (2001). Découverte du premier virus, le virus de la mosaïque du tabac: 1892 ou 1898? [Discovery of the first virus, the tobacco mosaic virus: 1892 or 1898?]. Comptes rendus de l'Academie des sciences. Serie III, Sciences de la vie, 324(10), 929–933. https://doi.org/10.1016/s0764-4469(01)01368-3

McWilliam, H., Li, W., Uludağ, M., Squizzato, S., Park, Y. M., Buso, N., Cowley, A. P., & López, R. (2013). Analysis Tool Web Services from the EMBL-EBI. Nucleic Acids Research, 41x(W1), W597–W600. https://doi.org/10.1093/nar/gkt376

Randhawa, G. S., Soltysiak, M. P., Roz, H. E., De Souza, C. P. E., Hill, K. A., & Kari, L. (2020). Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study. *PLOS ONE, 15(4), e0232391*. https://doi.org/10.1371/journal.pone.0232391

Remita, M. A., Halioui, A., Diouara, A. a. M., Daigle, B., Kiani, G., & Diallo, A. B. (2017). A machine learning approach for viral genome classification. *BMC Bioinformatics, 18(1)*. https://doi.org/10.1186/s12859-017-1602-3

Tang, X., Shang, J., & Sun, Y. (2022). RdRp-based sensitive taxonomic classification of RNA viruses for metagenomic data. *Briefings in Bioinformatics, 23(2)*. https://doi.org/10.1093/bib/bbac011