Cover Page

Title of Thesis

“Genome-based in silico models to predict infectious strains of Influenza A and Rotavirus A”

By Khushal Sharma

Under the Supervision of Prof. Dr. Gajendra P.S. Raghava

Indraprastha Institute of Information Technology Delhi May, 2022

Inner Cover Page

Inner Cover Page

Title of Thesis

"Genome-based in silico models to predict infectious strains of Influenza A and Rotavirus A"

By Khushal Sharma

Submitted

in partial fulfillment of the requirements for the degree of

Master of Technology

to

Indraprastha Institute of Information Technology Delhi May,

2022

Certificate

This is to certify that the thesis titled **“Genome-based in silico models to predict infectious strains of Influenza A and Rotavirus A”** being submitted by Khushal Sharma to the Indraprastha Institute of Information Technology Delhi, for the award of the Master of Technology, is an original research work carried out by her under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

May 2022

Prof.Gajendra P.S. Raghava
Department of Computational Biology
Indraprastha Institute of Information Technology Delhi
New Delhi 110 020

Acknowledgment

I want to take this opportunity to acknowledge my supervisor, Prof. Prof. Gajendra P S Raghava from Indraprastha Institute of Information and Technology, Delhi, for whom I have a deep sense of gratitude and indebtedness for giving me the wonderful opportunity to work under his guidance. I would also like to thank the Computational Biology Ph.D. scholars Anjali Dhall and Sumeet Patiyal for guiding through the course of my research. I would also like to express my gratitude to Trinita Roy, who collaborated with us in the making of the Protein module of FluSPred. Finally, I would like to thank all my friends and family who have provided invaluable support during my master's program.

Khushal Sharma
M.Tech CB

Table of Contents

List of Figures

List of Tables

Table 13: Model performance using machine learning models based on ALLCOMP(K = 1, 3) features on training and validation dataset

Abstract

The transmission of pathogens or infectious agents from animals to humans, i.e. zoonosis, is of global concern. Outbreaks pertaining to zoonosis have been the one of the major causes of fatalities worldwide. Both Influenza A, a viral disease that infects the lungs, and Rotavirus A, an enteric disease in humans and livestock, are zoonotic. Furthermore, recent outbreaks such as Ebolavirus, Zikavirus, Coronavirus are all examples of viruses of zoonotic origin, increasing in frequency due to globalization. This is where host reservoir prediction becomes essential as a preventive measure in order to curb a potential outbreak. In this study, we have made insilico machine learning based models for the prediction of host reservoir of Influenza A and Rotavirus A, whether the strain has the chances of being infectious to humans or not. This analysis would aid to the preventive measures healthcare authorities can take in order to control and mitigate possible outbreaks. We achieved 0.979 accuracy as well as area under the curve (AUC) on Influenza A based models for the training dataset. The accuracy and area under the curve (AUC) for the validation dataset is 0.98. Similarly, for Rotavirus A based models, training accuracy and area under the curve (AUC) is 0.986, validation accuracy is 0.953, and area under the curve (AUC) is 0.915. We have provided webservers FluSPred(https://webs.iiitd.edu.in/raghava/fluspred/), for Influenza A, and RotaVPred(https://webs.iiitd.edu.in/raghava/rotavpred/), for Rotavirus A, which have been made open source for the scientific community.

Chapter 1: Introduction

A zoonosis (zoonotic disease or zoonoses -plural) is a contagious disease transmitted between species from animals to humans (or from humans to animals)[1]. They are inherently transmissible from vertebrate animals to humans, constituting many existing diseases in humans.
The infectious disease is disseminated with the help of pathogens or infectious agents such as a bacterium, virus, parasite, or prions, capable of crossing the species barrier to infect humans [2].

The disease spreads when the first infected human spreads to other humans. Ebola, Rabies, Influenza, and SARS are some instances of diseases having a zoonotic origin[3]. Out of 1,415 pathogens known to infect humans, including 217 viruses and prions, 538 bacteria and rickettsia, 307 fungi, 66 protozoa and 287 helminths, 61%, i.e. 868 pathogens were zoonotic and 175 pathogens are considered to be 'emerging' [4]. Typically, most human diseases have an animal origin. However, diseases directly involved in non-human to human transmission via some infectious agent are known to be direct zoonoses[5]. Disease transmission applying an intermediate species called vectors, carrying the pathogens without getting sick, are known to be vector-borne zoonotic diseases[6]. On the other hand, when animals get infected by humans, it is known as reverse zoonosis[7]. Precarious animal viruses can replicate in human cells requiring only a few mutations, usually randomly arising in the natural reservoir.

There is evidence that viruses jump species barriers to affect animals and, in rare cases, humans[8]. As shown in Figure 1, different virus strains can affect humans and cause epidemics as the strains may have the potential to overcome species barriers and infect humans.

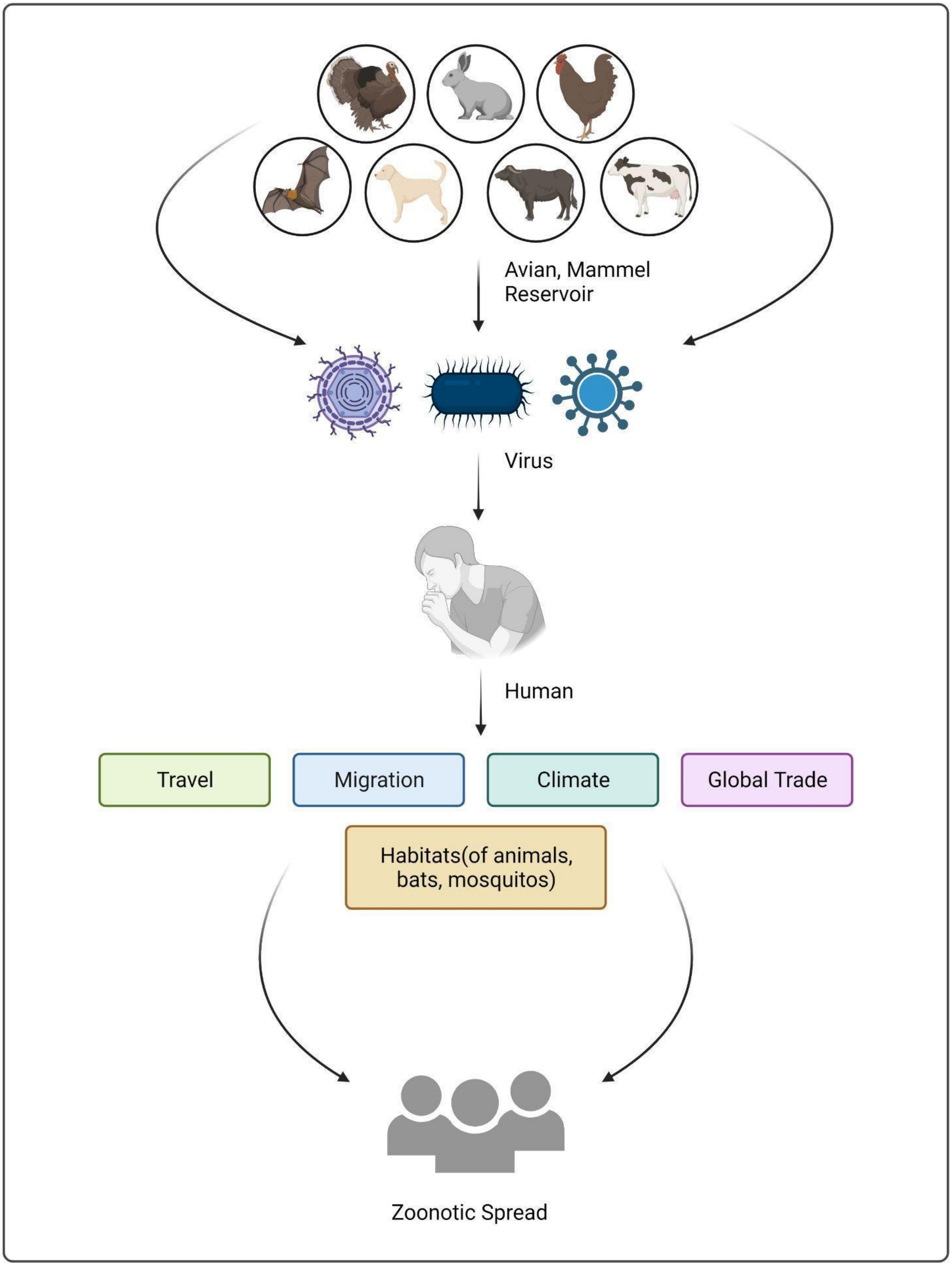

Avian, Mammel
Reservoir
Virus
Human
Travel
Migration
Climate
Global Trade
Habitats(of animals, bats, mosquitos)
Zoonotic Spread

Figure 1: Spread of a zoonotic virus from animal to humans further causing an outbreak among humans

Cross-species transmission of zoonotic pathogens is a substantial threat to humans and livestock[9]. Determining the origin of the pathogen during an outbreak becomes essential to aid in controlling and eradicating the disease[10]. The pathogen takes time to fully adapt to the human cells to be capable of spreading from human to human. If the host reservoir is known, the host species can be isolated or quarantined in order to limit the intensity of the outbreak[11].

Problem statement

We have made a methodical attempt to develop machine learning-based models to predict host reservoir of Influenza A virus and Rotavirus A. Influenza is a contagious viral disease, where there is a sudden rise in body temperature, headache, myalgia, lethargy, and dry cough. Influenza A virus occurs naturally among wild aquatic birds like geese, swans, waterfowl and it is responsible for causing avian influenza in birds, including domestic poultry [12]. Rotavirus A is an enteric disease in humans and livestock, notably in young calves and piglets. Group A rotaviruses (GARVs) account for up to 1 million children deaths each year, chiefly in developing countries[13]. This study aims to monitor infectious strains in non-human reservoirs to control infectious strains spreading from non-human to human hosts. Genome sequences of Influenza A virus and Rotavirus A were compiled. Models using different machine learning techniques were devised where features of proteins and genome were calculated using Nfeature. For the scientific community, we provide freely accessible web-servers named “FluSPred” for Influenza A and RotaVirusPred for Rotavirus A to predict infectious strains with the help of genome sequence using the best prediction models.

Influenza is a contagious viral disease, where there is a sudden rise in body temperature, headache, myalgia, lethargy, and dry cough. Influenza A virus occurs naturally among wild aquatic birds like geese, swans, waterfowl and it is responsible for causing avian influenza in birds, including domestic poultry [14]. Rotavirus A is an enteric disease in humans and livestock, notably in young calves and piglets. Group A rotaviruses (GARVs) account for up to 1 million children deaths each year, chiefly in developing countries[13].

Literature Review

Many researchers have made computational tools to predict the host of a virus using genomic sequences and some with protein sequences, physicochemical properties, environmental and phylogenetic analysis, genetics, evolution, and other public health factors. Supervised learning methods on viral contigs from metagenomic data to predict bacterial hosts of viruses was done by Zhang et al.. Viruses and viral contigs of the specific bacterial hosts were predicted by logistic regression, support vector machine, random forest, Gaussian naive Bayes, and Bernoulli naive Bayes techniques, taking "relative word frequencies" as the features of the virus, where random forest showed the best when k was taken as 6 [15]. Li and Sun also used support vector machine as well as alignment-based and alignment-free methods to predict the host of Influenza A, rabies, and coronavirus taking nucleotide frequencies as features [16].

Probabilistic methods such as WIsH is a tool that predicts prokaryotic hosts from the genomic contigs of phages [17]. In the study of Ahlgren et al, the host of a virus was predicted using Oligonucleotide frequency or ONF patterns, where the ONF similarity in the host was determined [18]. Along similar lines of probabilistic methods, the metagenomic study was conducted to bacteriophage-host relationships [19]. CRISPR sequences as well as sequence homology, alignment-free similarity measures were used to evaluate 1075 virus-host pairs resulting in 62% accuracy at genus and 85% accuracy at the phylum level, using Markov random field model for the tool VirHostMatcher [20]. In another study, biological, ecological, and life-history traits of rodents were analyzed, to predict the reservoir status of rodents, which are known to be carriers of various zoonotic pathogens, distinguishing reservoirs from non-reservoirs. Predicting hotspots of zoonotic diseases takes into account environmental factors, phylogenetic factors as well as human population present near the hotspot [21].

Zoonotic pathogens, once successfully making the species jump, need to adapt to the host environment in order to be able to circulate within that particular species. If the species is human then transmissibility is generally associated with low host mortality, non-segmented genomes, lack of insect vector, and chronic infection [22]. Along the lines, a gradient boosted regression model was used to predict specific virus species that, after zoonotic host tropism, have the potential for human-to-human transmission [23]. Virus Deep learning HOst Prediction or VIDHOP is a Deep Learning-based tool for predicting the host of viruses using the genome sequence of influenza A, rabies lyssavirus, rotavirus A. They showed that deep neural networks are effective for predicting the host with 100bp to 400bp length and a highly unbalanced dataset [24].

In our study, we aim to make a web-based tool for the prediction of novel strains of Influenza A virus, whether it is capable of infecting humans or non-human hosts, using genomic sequences. Individual datasets for the genomic sequences were made and models were computed based on them. For genome sequences, machine learning based models were made where binary profiling as well as models where CDK(k-mer composition), and RDK with k-mers of 1, 2, and 3 were used for feature generation.

These studies show that genome sequences are very efficient for the determining the host tropism of zoonotic pathogens and changes take place at the molecular level for the virus being capable of crossing the species barrier.

Chapter 2: Host Prediction of Influenza A

Workflow

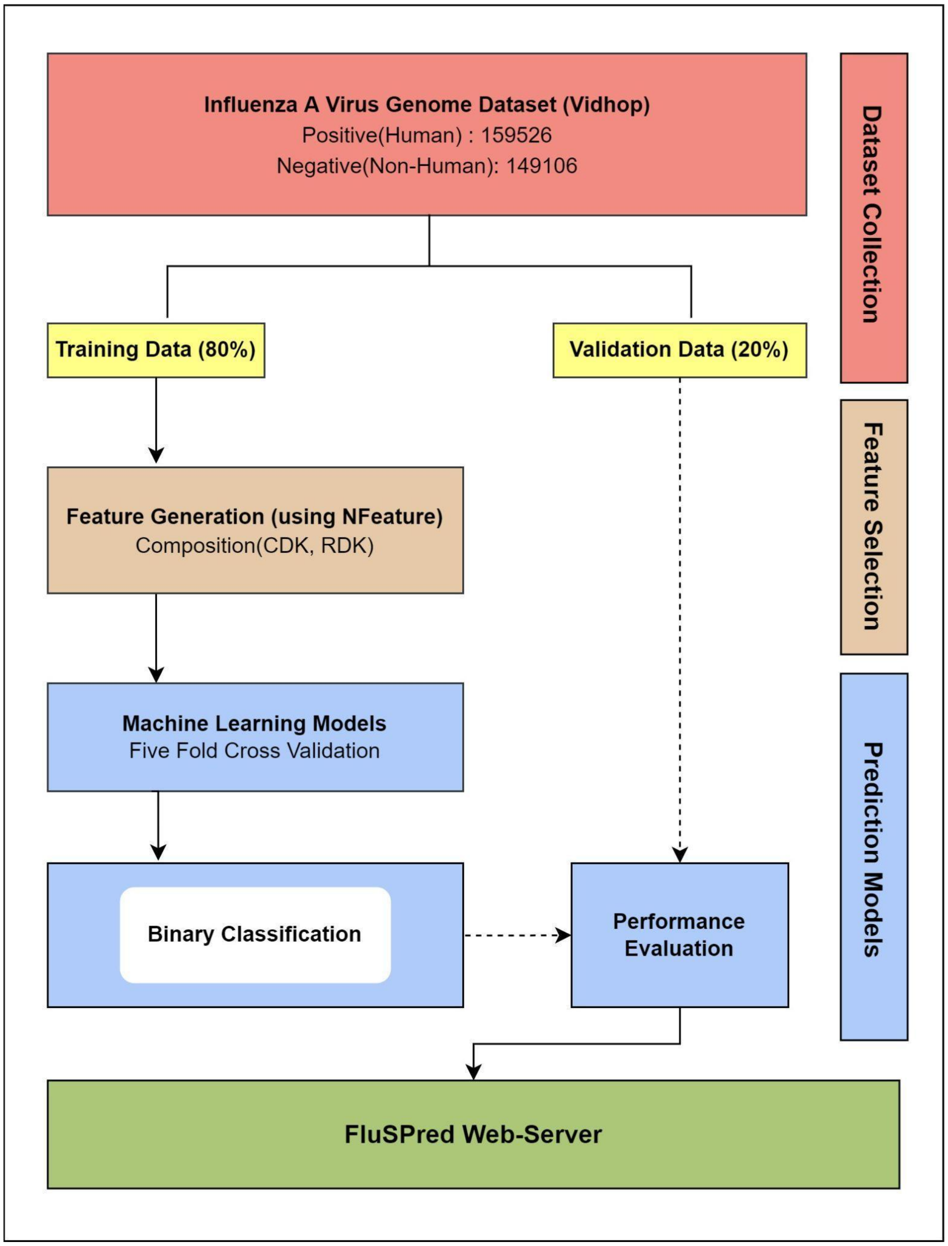


Figure 2: Flow diagram of the FluSpread Project, From collecting data to Showing Results on webpage

Material and Methods

Dataset Preparation

The dataset for whole-genome sequences for Influenza A was acquired from the research paper VIDHOP, a study conducted by Mock et al. [24]. The dataset contained 312617 sequences, out of which 159526 were taken from humans. This was regarded as a "positive" dataset. 153091 sequences were derived from other animals considered as "negative" datasets. From the negative dataset, 104 sequences belonging to *Leistomus xantharus* and 91 sequences of *Plantago princeps* were removed as only avian and mammalian host sequences were being considered. Apart from this, the duplicate sequences were removed, and finally, a total of 308632 sequences were taken, with 159526 positive and 149106 negative sequences. The positive and the negative sequences were assigned values of 1 and 0 respectively for binary classification. The total number of samples was divided into training and validation data in 80 : 20 ratio, for training and external validation[25, 26]

Feature Generation

Feature extraction for genome sequences was done using Nfeature, a wide-ranging, feature-rich, user-friendly package to calculate a variety of features of Nucleic Acids such as Deoxyribonucleic Acids (DNA) and Ribonucleic Acids (RNA), and perform complex classification tasks and feature calculation tasks with ease[27]. Composition-based features such as the composition of DNA or RNA sequence for all K-Mers or CDK and Reverse complement of DNA for all K-mers or RDK were computed.

CDK is a feature extraction method that calculates the composition of all 4 nucleotides. Here K could be 1, 2, or 3.

$$CDKi = Ni / K \qquad \text{(i)}$$

where i is the ith nucleotide or k-mer of nucleotide and L is the length of the input sequence.

For k=2 length of the sequence becomes (L-1) and for k=3, the length of the input sequence will be (L-3).

RDK is a feature extraction method that calculates reverse complement K-Mer composition of nucleotides.

Machine Learning Models

Several machine learning algorithms have been executed in this analysis for developing prediction models that include Random Forest (RF), K-Nearest Neighbours (KNN), Decision Tree (DT), Support Vector Machine (SVM), XGBoost (XGB), Gaussian Naive Bayes (GNB). KNN is an instance-based classification method[28] that classifies based on the vote of the nearest neighbor data points, and it holds the instances of the training variables. The majority vote of the nearest neighbor of each data point decides the classification. RF is an ensemble-based method for classification, which constructs many decision trees[29]. It fits a considerable amount of DTs to predict the response variable as an individual tree while training. DT is a non-parametric-based supervised learning algorithm. It predicts the response variable by learning the decision rules from the data features[30]. GBM algorithm is a probabilistic classification approach based on the Bayes theorem. This algorithm assumes that the continuous variables of each class follow the normal or Gaussian distribution. XGB is an ensemble method that implements the gradient boosted decision trees designed for speed and performance [31]. SVM is a supervised learning algorithm that finds a hyperplane in an N-dimensional space for the classification [32]. For genome sequences, DT, KNN, XGB, GNB, SVM, and RF-based models were generated using CDK and RDK. The model was then fine-tuned on the training data by taking different thresholds for probabilities as 0.8,0.5, 0.4, 0.2, 0.1. These classification techniques were implemented using python-library sci-kit learn [33].

Cross-Validation

5-fold cross-validation was performed on each model after the data were split into training and validation datasets. Additionally, the training data was separated into training and testing datasets during the five-fold cross-validation process. The entire training data gets divided into five equivalent folds in this approach. For each iteration, four folds are used for training, one fold is used as the testing dataset, and training takes place. The whole procedure gets iterated five times, where every fold gets a chance to be used as testing data. The mean of the results for each fold of the cross-fold validation was taken for the training dataset. This is a standard procedure used in many studies[34-37].

Evaluation Parameters

In the present analysis, we have used the definitive performance evaluation parameters such as accuracy, sensitivity, specificity, threshold-dependent parameters, and area under the curve (AUC), an independent threshold parameter devised by sensitivity vs. 1-specificity. Along with this, precision, recall, as well as Matthews Correlation Coefficient (MCC) were calculated by:

$$Sensitivity = \frac{T_P}{T_P+F_N} \qquad (ii)$$

$$Specificity = \frac{T_N}{T_N + F_P} \qquad (iii)$$

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \qquad (iv)$$

$$MCC = \frac{\left(T_P * T_N\right) - \left(F_P * F_N\right)}{\sqrt{(T_P + F_P)(T_P + F_N)(T_N + F_P)(T_N + F_N)}} \qquad (v)$$

Where T_P is true positive, T_N is true negative, F_P is false positive and F_N is false negative.

Results

Performance of Machine Learning Models using CDK Features

Genome-based classification models were developed using various machine learning techniques such as RF, KNN, SVM, DT, NB, and XGB. Here, RDK and CDK methods from Nfeature were used to compute the compositional based features for nucleotide sequences. which were used to develop prediction models for finding the human/non-human infectious strains. Likewise, we have generated CDK based features, where the random forest of threshold 0.5 achieved the highest accuracy (97.6%) and AUC of 0.97 on the training data and 98% accuracy and AUC of 0.98 on the validation data, as shown in the table below (Table 5). Here, KNN showed almost similar results with an accuracy of 97.6% and AUC of 0.975 on training data and 97.7% accuracy, and AUC of 0.977 on the validation dataset. The results for CDK are provided in the table below.

Table 1: Model performance using machine learning models based on CDK (K = 1, 2, 3) features on training and validation dataset on threshold 0.1

		Training					Validation				
Model	k-mer	Accuracy	Sensitivity	Specificity	*AUC*	MCC	Accuracy	Sensitivity	Specificity	*AUC*	MCC
	1	0.802	0.944	0.65	0.797	0.626	0.805	0.946	0.654	0.8	0.632
	2	0.956	0.991	0.919	0.955	0.915	0.956	0.992	0.918	0.955	0.915

KNN

	3	0.965	0.994	0.934	0.964	0.932	0.966	0.994	0.937	0.965	0.934
RF	1	0.81	0.98	0.629	0.804	0.656	0.813	0.98	0.633	0.806	0.66
	2	0.913	0.995	0.825	0.91	0.836	0.915	0.996	0.828	0.912	0.84
	3	0.933	0.996	0.866	0.931	0.874	0.934	0.995	0.868	0.932	0.874
SVM	1	0.515	1	0.00E+00	0.5	0.004	0.518	1	0.00E+00	0.5	0.004
	2	0.531	0.995	0.037	0.516	0.114	0.532	0.995	0.038	0.516	0.116
	3	0.662	0.983	0.315	0.649	0.406	0.66	0.984	0.314	0.649	0.407
DT	1	0.859	0.886	0.83	0.858	0.719	0.859	0.887	0.83	0.503	0.006
	2	0.944	0.948	0.939	0.943	0.888	0.942	0.947	0.937	0.502	0.004
	3	0.957	0.959	0.955	0.957	0.914	0.955	0.96	0.951	0.955	0.911
NB	1	0.55	0.997	0.074	0.535	0.188	0.549	0.997	0.07	0.534	0.184
	2	0.561	0.808	0.297	0.552	0.123	0.56	0.807	0.295	0.551	0.12
	3	0.617	0.771	0.451	0.611	0.236	0.616	0.771	0.449	0.61	0.234
XGB	1	0.585	0.996	0.147	0.571	0.276	0.584	0.996	0.144	0.57	0.272
	2	0.649	0.994	0.282	0.638	0.399	0.651	0.994	0.282	0.638	0.4
	3	0.737	0.991	0.464	0.728	0.543	0.736	0.991	0.463	0.727	0.541

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #NB: Naive bayes, XGB: XGBoost

Table 2: Model performance using machine learning models based on CDK (K = 1, 2, 3) features on training and validation dataset on threshold 0.2

		Training					Validation				
Model	**k-mer**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**
KNN	1	0.802	0.944	0.65	0.797	0.626	0.805	0.946	0.654	0.8	0.632
	2	0.956	0.991	0.919	0.955	0.915	0.956	0.992	0.918	0.955	0.915
	3	0.965	0.994	0.934	0.964	0.932	0.966	0.994	0.937	0.965	0.934
RF	1	0.854	0.963	0.738	0.851	0.724	0.856	0.963	0.741	0.852	0.727
	2	0.953	0.99	0.914	0.952	0.909	0.954	0.991	0.913	0.952	0.91
	3	0.963	0.992	0.931	0.962	0.928	0.965	0.993	0.936	0.964	0.932
SVM	1	0.516	0.999	0.001	0.5	0.023	0.519	1	0.001	0.5	0.025
	2	0.585	0.983	0.161	0.572	0.256	0.585	0.983	0.159	0.571	0.254
	3	0.741	0.964	0.499	0.732	0.529	0.739	0.965	0.498	0.731	0.528
DT	1	0.86	0.885	0.833	0.859	0.721	0.86	0.886	0.833	0.503	0.007
	2	0.944	0.948	0.939	0.943	0.888	0.942	0.947	0.937	0.502	0.004

	3	0.957	0.959	0.955	0.957	0.914	0.955	0.958	0.952	0.5	0.001
NB	1	0.565	0.935	0.17	0.552	0.165	0.564	0.938	0.165	0.552	0.166
	2	0.569	0.784	0.34	0.562	0.139	0.569	0.782	0.339	0.561	0.137
	3	0.643	0.768	0.509	0.638	0.288	0.643	0.769	0.507	0.638	0.287
XGB	1	0.689	0.981	0.379	0.68	0.456	0.685	0.979	0.372	0.675	0.446
	2	0.777	0.983	0.557	0.77	0.604	0.777	0.982	0.557	0.77	0.602
	3	0.868	0.981	0.748	0.864	0.754	0.866	0.98	0.745	0.862	0.751

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #NB: Naive bayes, XGB: XGBoost

Table 3: Model performance using machine learning models based on CDK (K = 1, 2, 3) features on training and validation dataset on threshold 0.4

		Training					Validation				
Model	**k-mer**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**
KNN	1	0.841	0.854	0.828	0.841	0.682	0.843	0.856	0.828	0.842	0.685
	2	0.971	0.982	0.96	0.971	0.943	0.971	0.982	0.959	0.971	0.943
	3	0.976	0.987	0.964	0.975	0.952	0.977	0.987	0.966	0.977	0.955
RF	1	0.887	0.922	0.849	0.886	0.775	0.887	0.921	0.849	0.885	0.774
	2	0.972	0.979	0.964	0.971	0.944	0.972	0.979	0.963	0.971	0.944
	3	0.977	0.985	0.97	0.977	0.955	0.979	0.986	0.972	0.979	0.959
SVM	1	0.558	0.815	0.285	0.55	0.119	0.563	0.818	0.289	0.553	0.127
	2	0.658	0.849	0.454	0.652	0.332	0.656	0.845	0.454	0.649	0.327
	3	0.805	0.896	0.707	0.801	0.617	0.806	0.898	0.708	0.803	0.619
DT	1	0.862	0.88	0.842	0.861	0.724	0.862	0.88	0.842	0.503	0.007
	2	0.944	0.948	0.939	0.943	0.888	0.942	0.947	0.938	0.502	0.004
	3	0.957	0.959	0.955	0.957	0.914	0.955	0.958	0.952	0.5	0.001
NB	1	0.56	0.787	0.318	0.553	0.12	0.564	0.796	0.316	0.556	0.128
	2	0.601	0.759	0.432	0.595	0.203	0.598	0.757	0.427	0.592	0.196
	3	0.67	0.754	0.58	0.667	0.34	0.669	0.755	0.577	0.666	0.339
XGB	1	0.755	0.91	0.589	0.75	0.531	0.748	0.904	0.582	0.743	0.517
	2	0.883	0.952	0.809	0.881	0.772	0.881	0.951	0.807	0.879	0.769
	3	0.94	0.969	0.908	0.939	0.881	0.937	0.967	0.905	0.936	0.876

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #NB: Naive bayes, XGB: XGBoost

Table 4: Model performance using machine learning models based on CDK (K = 1, 2, 3) features on training and validation dataset on threshold 0.5

		Training					Validation				
Model	**k-mer**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**
KNN	1	0.841	0.854	0.828	0.841	0.682	0.843	0.856	0.828	0.842	0.685
	2	0.971	0.982	0.96	0.971	0.943	0.971	0.982	0.959	0.971	0.943
	3	0.976	0.987	0.964	0.975	0.952	0.977	0.987	0.966	0.977	0.955
RF	1	0.89	0.894	0.886	0.89	0.78	0.889	0.892	0.886	0.889	0.779
	2	0.972	0.971	0.974	0.972	0.945	0.973	0.972	0.973	0.973	0.946
	3	0.979	0.98	0.978	0.979	0.958	0.98	0.981	0.98	0.98	0.961
SVM	1	0.606	0.71	0.494	0.602	0.21	0.611	0.711	0.502	0.607	0.219
	2	0.687	0.719	0.652	0.686	0.373	0.686	0.717	0.652	0.685	0.371
	3	0.817	0.853	0.779	0.816	0.635	0.82	0.856	0.782	0.819	0.641
DT	1	0.861	0.869	0.853	0.861	0.722	0.862	0.869	0.854	0.502	0.005
	2	0.943	0.947	0.939	0.943	0.887	0.942	0.946	0.938	0.502	0.004
	3	0.957	0.959	0.955	0.957	0.914	0.955	0.958	0.953	0.5	0.001
NB	1	0.598	0.739	0.448	0.593	0.196	0.601	0.747	0.446	0.596	0.203
	2	0.618	0.739	0.489	0.614	0.236	0.615	0.737	0.483	0.61	0.229
	3	0.678	0.742	0.609	0.676	0.356	0.676	0.743	0.604	0.673	0.351
XGB	1	0.78	0.846	0.71	0.778	0.563	0.775	0.841	0.704	0.773	0.552
	2	0.9	0.922	0.877	0.9	0.801	0.899	0.92	0.878	0.899	0.799
	3	0.947	0.956	0.937	0.946	0.894	0.946	0.955	0.936	0.946	0.892

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #NB: Naive bayes, XGB: XGBoost

Table 5: Model performance using machine learning models based on CDK (K = 1, 2, 3) features on training and validation dataset on threshold 0.8

		Training					Validation				
Model	**k-mer**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**
KNN	1	0.8	0.68	0.927	0.804	0.624	0.799	0.679	0.929	0.804	0.624
	2	0.966	0.953	0.981	0.967	0.933	0.964	0.95	0.979	0.965	0.929
	3	0.973	0.966	0.981	0.974	0.947	0.975	0.967	0.983	0.975	0.95
RF	1	0.852	0.751	0.961	0.856	0.725	0.852	0.75	0.962	0.856	0.725
	2	0.946	0.905	0.99	0.947	0.896	0.947	0.907	0.99	0.948	0.898
	3	0.962	0.935	0.992	0.963	0.926	0.963	0.935	0.993	0.964	0.928

SVM	1	0.484	0	0.999	0.499	-0.001	0.481	0.00E+00	0.999	0.499	0
	2	0.528	0.121	0.963	0.542	0.156	0.528	0.12	0.964	0.542	0.156
	3	0.7	0.48	0.937	0.709	0.465	0.705	0.486	0.939	0.712	0.473
DT	1	0.852	0.841	0.863	0.852	0.704	0.852	0.841	0.863	0.503	0.006
	2	0.943	0.947	0.939	0.943	0.887	0.942	0.945	0.938	0.502	0.004
	3	0.957	0.959	0.955	0.957	0.914	0.955	0.958	0.953	0.5	0.001
NB	1	0.481	0.001	0.993	0.497	-0.038	0.481	0.001	0.992	0.497	-0.038
	2	0.654	0.602	0.71	0.656	0.314	0.652	0.6	0.708	0.654	0.31
	3	0.699	0.704	0.693	0.698	0.397	0.694	0.7	0.686	0.693	0.387
XGB	1	0.641	0.333	0.97	0.652	0.39	0.639	0.329	0.97	0.65	0.387
	2	0.791	0.607	0.987	0.797	0.637	0.786	0.598	0.987	0.792	0.629
	3	0.878	0.779	0.985	0.882	0.777	0.876	0.776	0.984	0.88	0.773

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #NB: Naive bayes, XGB: XGBoost

Performance of Machine Learning Models using RDK Features

RDK based features achieve maximum accuracy of 97.6% and AUC of 0.976 on the training data and validation data using RF-based classifiers. Similarly, KNN based models also show comparable results on both training and validation dataset (Table 6). However, XGB performs quite less with an accuracy of 92.5% and an AUC of 0.925 on validation datasets. Results of SVM, DT, and NB classifiers could not perform well on the training as well as validation dataset, as given in Table 6. The complete results are provided in the table below

Table 6: Model performance using machine learning models based on RDK(K = 1, 2, 3) features on training and validation dataset for threshold 0.1

		Training					Validation				
Model	**k-mer**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**
KNN	1	0.933	0.987	0.875	0.931	0.871	0.933	0.987	0.875	0.931	0.871
	2	0.933	0.987	0.875	0.931	0.871	0.933	0.987	0.875	0.931	0.871
	3	0.964	0.993	0.932	0.962	0.929	0.963	0.993	0.931	0.962	0.928

RF	1	0.886	0.993	0.772	0.882	0.789	0.888	0.993	0.774	0.884	0.792
	2	0.886	0.993	0.772	0.882	0.789	0.888	0.993	0.774	0.884	0.792
	3	0.924	0.995	0.847	0.921	0.856	0.923	0.996	0.847	0.921	0.856
SVM	1	-	-	-	-	-	-	-	-	-	-
	2	0.533	0.997	0.037	0.517	0.126	0.533	0.996	0.037	0.516	0.122
	3	0.576	0.985	0.138	0.562	0.236	0.574	0.986	0.136	0.561	0.235
DT	1	0.584	0.239	0.951	0.595	0.269	0.712	0.939	0.466	0.496	-0.007
	2	0.923	0.929	0.916	0.923	0.846	0.926	0.932	0.919	0.502	0.005
	3	0.949	0.954	0.944	0.949	0.898	0.949	0.951	0.946	0.502	0.004
NB	1	0.56	0.939	0.157	0.548	0.156	0.564	0.941	0.16	0.55	0.162
	2	0.566	0.827	0.287	0.557	0.136	0.567	0.828	0.285	0.556	0.135
	3	0.582	0.793	0.357	0.575	0.167	0.58	0.788	0.357	0.572	0.161
XGB	1	0.537	0.998	0.045	0.522	0.148	0.541	0.998	0.048	0.523	0.154
	2	0.624	0.994	0.228	0.611	0.351	0.623	0.994	0.223	0.608	0.347
	3	0.673	0.993	0.329	0.661	0.437	0.675	0.992	0.335	0.663	0.44

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #NB: Naive bayes, XGB: XGBoost

Table 7: Model performance using machine learning models based on RDK(K = 1, 2, 3) features on training and validation dataset for threshold 0.2

		Training					**Validation**				
Model	**k-mer**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**
KNN	1	0.933	0.987	0.875	0.931	0.871	0.933	0.987	0.875	0.931	0.871
	2	0.933	0.987	0.875	0.931	0.871	0.933	0.987	0.875	0.931	0.871
	3	0.964	0.993	0.932	0.962	0.929	0.963	0.993	0.931	0.962	0.928
RF	1	0.934	0.986	0.878	0.932	0.872	0.934	0.987	0.878	0.933	0.874
	2	0.934	0.986	0.878	0.932	0.872	0.934	0.987	0.878	0.933	0.874
	3	0.96	0.992	0.926	0.959	0.922	0.961	0.992	0.928	0.96	0.924
SVM	1	-	-	-	-	-	-	-	-	-	-
	2	0.564	0.989	0.11	0.549	0.211	0.565	0.988	0.111	0.549	0.209
	3	0.631	0.957	0.281	0.619	0.327	0.629	0.956	0.281	0.618	0.324
DT	1	0.749	0.924	0.563	0.744	0.526	0.747	0.922	0.556	0.496	-0.007
	2	0.923	0.929	0.916	0.923	0.846	0.926	0.932	0.919	0.502	0.005
	3	0.949	0.954	0.944	0.949	0.898	0.949	0.951	0.946	0.502	0.004
NB	1	0.547	0.883	0.188	0.536	0.1	0.552	0.886	0.191	0.539	0.109

	2	0.566	0.795	0.321	0.558	0.132	0.566	0.795	0.318	0.557	0.13
	3	0.6	0.782	0.405	0.593	0.203	0.597	0.777	0.404	0.59	0.196
XGB	1	0.55	0.994	0.077	0.536	0.182	0.553	0.995	0.077	0.536	0.185
	2	0.734	0.977	0.474	0.726	0.528	0.736	0.978	0.475	0.726	0.53
	3	0.815	0.981	0.636	0.809	0.663	0.814	0.979	0.638	0.809	0.662

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #NB: Naive bayes, XGB: XGBoost

Table 8: Model performance using machine learning models based on RDK(K = 1, 2, 3) features on training and validation dataset for threshold 0.4

		Training					Validation				
Model	**k-mer**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**
KNN	1	0.956	0.969	0.941	0.955	0.912	0.956	0.97	0.941	0.956	0.913
	2	0.956	0.969	0.941	0.955	0.912	0.956	0.97	0.941	0.956	0.913
	3	0.975	0.986	0.963	0.974	0.95	0.975	0.986	0.963	0.974	0.95
RF	1	0.959	0.969	0.948	0.958	0.918	0.96	0.971	0.948	0.959	0.92
	2	0.959	0.969	0.948	0.958	0.918	0.96	0.971	0.948	0.959	0.92
	3	0.975	0.983	0.967	0.975	0.951	0.975	0.983	0.967	0.975	0.951
SVM	1	-	-	-	-	-	-	-	-	-	-
	2	0.64	0.873	0.391	0.632	0.304	0.642	0.872	3.95E-01	0.634	0.306
	3	0.725	0.866	0.573	0.72	0.462	0.728	0.869	0.578	0.723	0.469
DT	1	0.78	0.862	0.693	0.778	0.565	0.779	0.862	0.689	0.498	-0.003
	2	0.923	0.929	0.916	0.923	0.846	0.926	0.932	0.92	0.502	0.005
	3	0.949	0.954	0.944	0.949	0.898	0.949	0.951	0.946	0.502	0.004
NB	1	0.559	0.822	0.279	0.55	0.121	0.561	0.823	0.28	0.551	0.123
	2	0.595	0.774	0.403	0.588	0.191	0.596	0.775	0.402	0.589	0.192
	3	0.625	0.749	0.492	0.62	0.25	0.624	0.745	0.495	0.62	0.249
XGB	1	0.66	0.828	0.481	0.655	0.332	0.662	0.832	0.478	0.655	0.333
	2	0.849	0.925	0.769	0.847	0.705	0.851	0.926	0.771	0.848	0.709
	3	0.912	0.96	0.862	0.911	0.828	0.913	0.957	0.865	0.911	0.829

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #NB: Naive bayes, XGB: XGBoost

Table 9: Model performance using machine learning models based on RDK(K = 1, 2, 3) features on training and validation dataset for threshold 0.5

		Training					Validation				
Model	k-mer	Accuracy	Sensitivity	Specificity	*AUC*	MCC	Accuracy	Sensitivity	Specificity	*AUC*	MCC
KNN	1	0.956	0.969	0.941	0.955	0.912	0.956	0.97	0.941	0.956	0.913
	2	0.956	0.969	0.941	0.955	0.912	0.956	0.97	0.941	0.956	0.913
	3	0.975	0.986	0.963	0.974	0.95	0.975	0.986	0.963	0.974	0.95
RF	1	0.96	0.957	0.963	0.96	0.92	0.961	0.959	0.963	0.961	0.922
	2	0.96	0.957	0.963	0.96	0.92	0.961	0.959	0.963	0.961	0.922
	3	0.976	0.977	0.976	0.976	0.953	0.977	0.977	0.976	0.977	0.954
SVM	1	-	-	-	-	-	-	-	-	-	-
	2	0.63	0.656	0.602	0.629	0.259	0.632	0.656	0.606	0.631	0.263
	3	0.753	0.796	0.707	0.751	0.506	0.753	0.797	0.707	0.752	0.507
DT	1	0.783	0.806	0.76	0.783	0.567	0.783	0.8	0.765	0.498	-0.003
	2	0.923	0.928	0.917	0.922	0.845	0.925	0.93	0.92	0.502	0.004
	3	0.949	0.953	0.944	0.949	0.898	0.949	0.951	0.946	0.502	0.004
NB	1	0.588	0.778	0.386	0.582	0.179	0.588	0.777	0.385	0.581	0.176
	2	0.609	0.75	0.457	0.604	0.218	0.61	0.752	0.457	0.604	0.219
	3	0.64	0.74	0.533	0.637	0.281	0.639	0.736	0.536	0.636	0.278
XGB	1	0.671	0.748	0.588	0.668	0.342	0.669	0.75	0.582	0.666	0.338
	2	0.864	0.879	0.849	0.864	0.729	0.866	0.88	0.851	0.866	0.732
	3	0.926	0.938	0.912	0.925	0.852	0.925	0.936	0.913	0.925	0.85

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #NB: Naive bayes, XGB: XGBoost

Table 10: Model performance using machine learning models based on RDK(K = 1, 2, 3) features on training and validation dataset for threshold 0.8

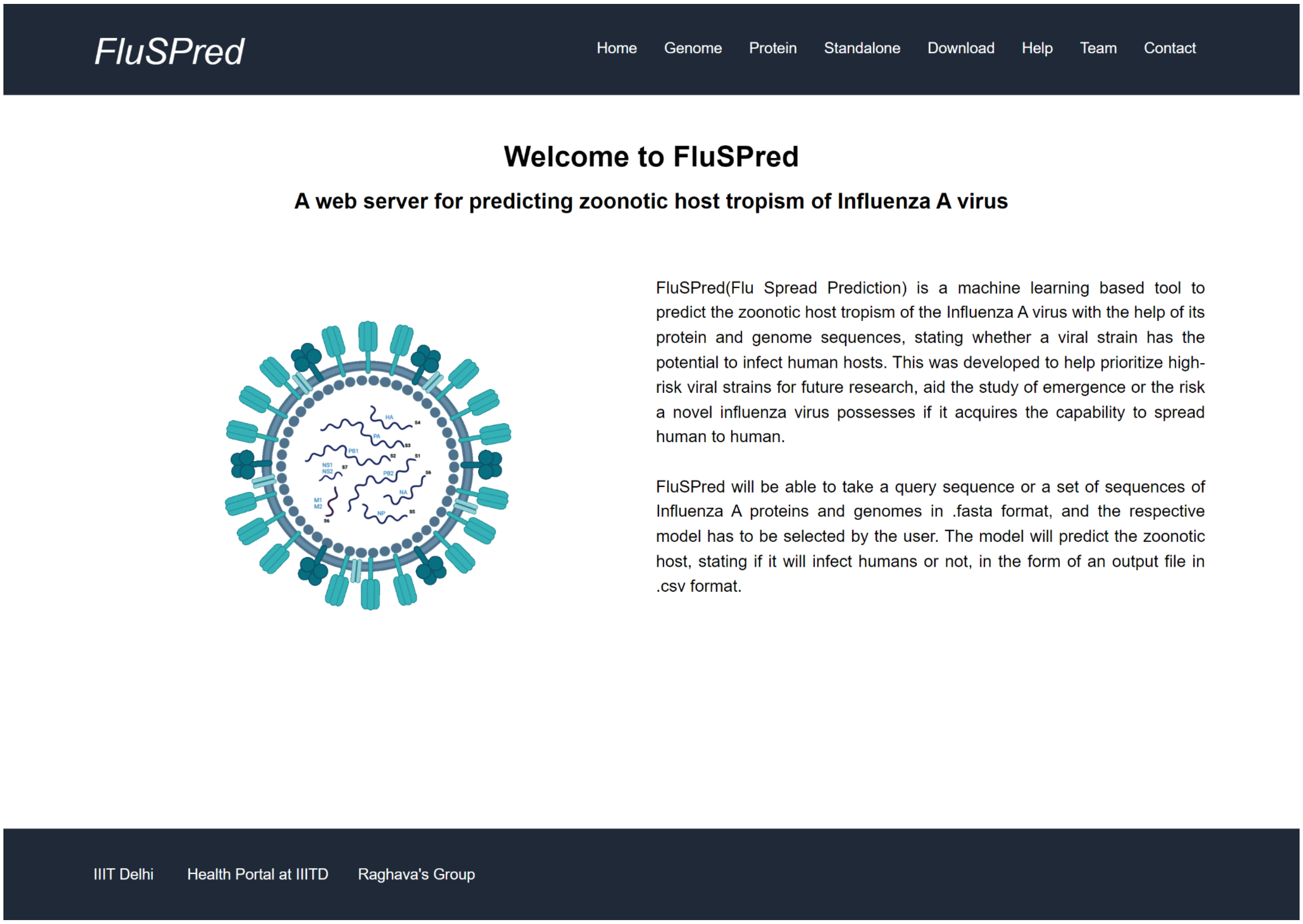
		Training					Validation				
Model	k-mer	Accuracy	Sensitivity	Specificity	*AUC*	MCC	Accuracy	Sensitivity	Specificity	*AUC*	MCC
KNN	1	0.945	0.919	0.974	0.946	0.893	0.945	0.919	0.973	0.946	0.892
	2	0.945	0.919	0.974	0.946	0.893	0.945	0.919	0.973	0.946	0.892
	3	0.971	0.961	0.982	0.971	0.942	0.972	0.963	0.982	0.972	0.945
RF	1	0.929	0.875	0.987	0.931	0.864	0.929	0.874	0.988	0.931	0.865
	2	0.929	0.875	0.987	0.931	0.864	0.929	0.874	0.988	0.931	0.865
	3	0.956	0.922	0.992	0.957	0.914	0.956	0.922	0.992	0.957	0.915
SVM	1	-	-	-	-	-	-	-	-	-	-

	2	0.533	0.118	0.976	0.547	0.183	0.529	0.111	0.977	0.544	0.177
	3	0.582	0.238	0.951	0.594	0.267	0.584	0.239	0.951	0.595	0.269
DT	1	0.718	0.541	0.906	0.723	0.478	0.713	0.531	0.91	0.499	0
	2	0.922	0.926	0.917	0.922	0.844	0.925	0.928	0.921	0.502	0.004
	3	0.949	0.953	0.944	0.949	0.898	0.949	0.951	0.946	0.502	0.004
NB	1	0.484	0	1	0.5	0	0.481	0	1	0.5	0
	2	0.656	0.615	0.699	0.657	0.315	0.655	0.616	0.697	0.657	0.314
	3	0.666	0.668	0.664	0.666	0.332	0.664	0.663	0.665	0.664	0.329
XGB	1	0.484	0	1	0.5	0	0.481	0	1	0.5	0
	2	0.762	0.565	0.972	0.769	0.583	0.76	0.563	0.973	0.768	0.582
	3	0.83	0.686	0.984	0.835	0.697	0.83	0.685	0.984	0.835	0.697

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #NB: Naive bayes, XGB: XGBoost

Architecture of the Web-Server

A web-server "FluSPred" (https://webs.iiitd.edu.in/raghava/fluspred/) as well as a standalone (https://github.com/raghavagps/fluspred) was developed incorporating the random forest model using CDK features for k=3, which was considered the best model based on the AUC results.



Welcome to FluSPred

A web server for predicting zoonotic host tropism of Influenza A virus

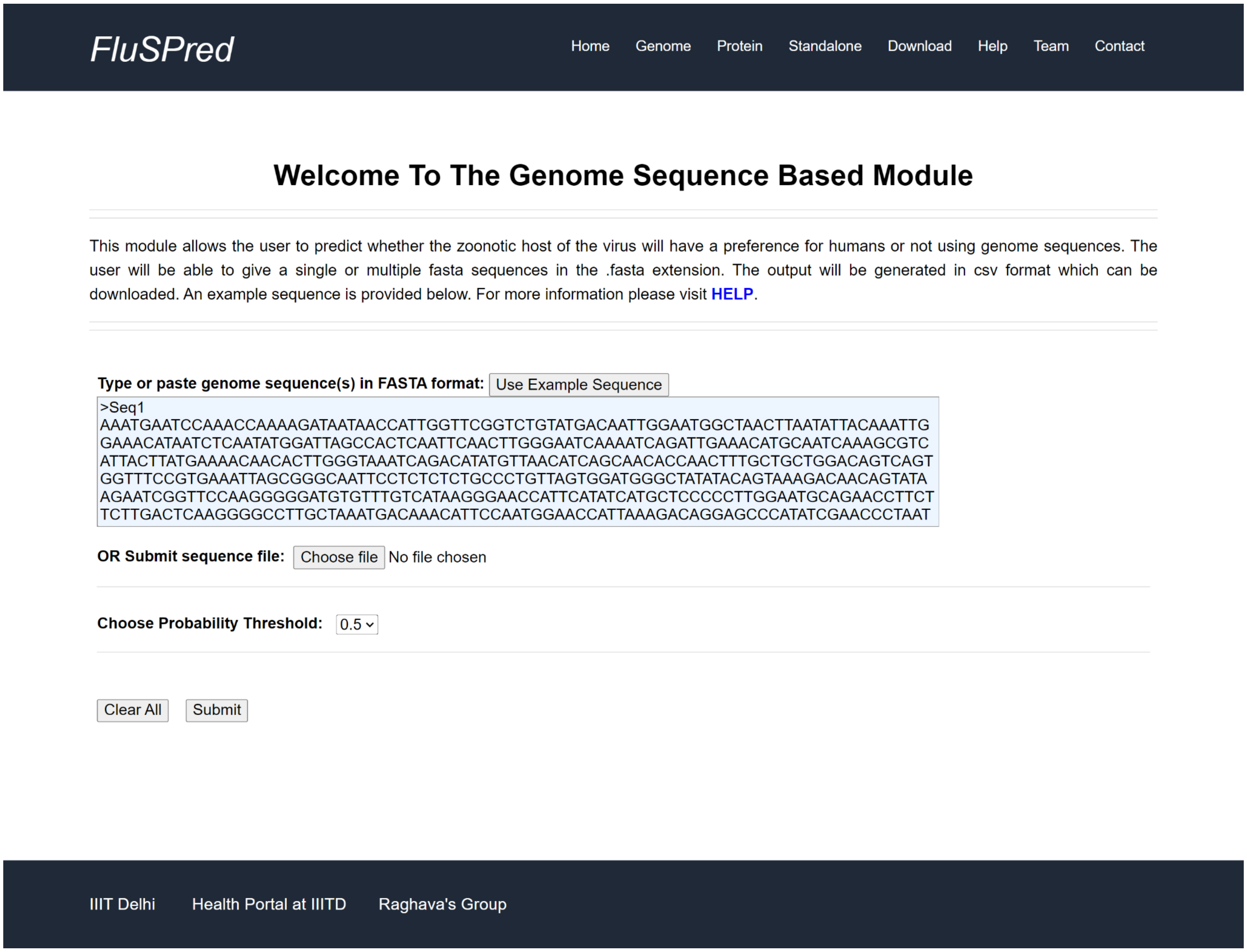
FluSPred(Flu Spread Prediction) is a machine learning based tool to predict the zoonotic host tropism of the Influenza A virus with the help of its protein and genome sequences, stating whether a viral strain has the potential to infect human hosts. This was developed to help prioritize high-risk viral strains for future research, aid the study of emergence or the risk a novel influenza virus possesses if it acquires the capability to spread human to human.

FluSPred will be able to take a query sequence or a set of sequences of Influenza A proteins and genomes in .fasta format, and the respective model has to be selected by the user. The model will predict the zoonotic host, stating if it will infect humans or not, in the form of an output file in .csv format.

Figure 3: FluSPred Web Server HomePage

Web Server was built in two parts, Front End and Back End. HTML & CSS were used for styling and VanillaJS for form validation and sequence submission logic for the front end. The Backend used Python and PHP to run the prediction model and generate a result file. The website was made keeping the focus on mobile and laptop users, and this is why it is fully responsive and functional with all screen sizes.

The Genome module contains the best model for the prediction of human and non-human infectious strains using genomic sequences. A single or multiple input sequences the standard .FASTA format must be provided and then the user has to click the “Submit” button.



Welcome To The Genome Sequence Based Module

This module allows the user to predict whether the zoonotic host of the virus will have a preference for humans or not using genome sequences. The user will be able to give a single or multiple fasta sequences in the .fasta extension. The output will be generated in csv format which can be downloaded. An example sequence is provided below. For more information please visit **HELP**.

Type or paste genome sequence(s) in FASTA format: Use Example Sequence

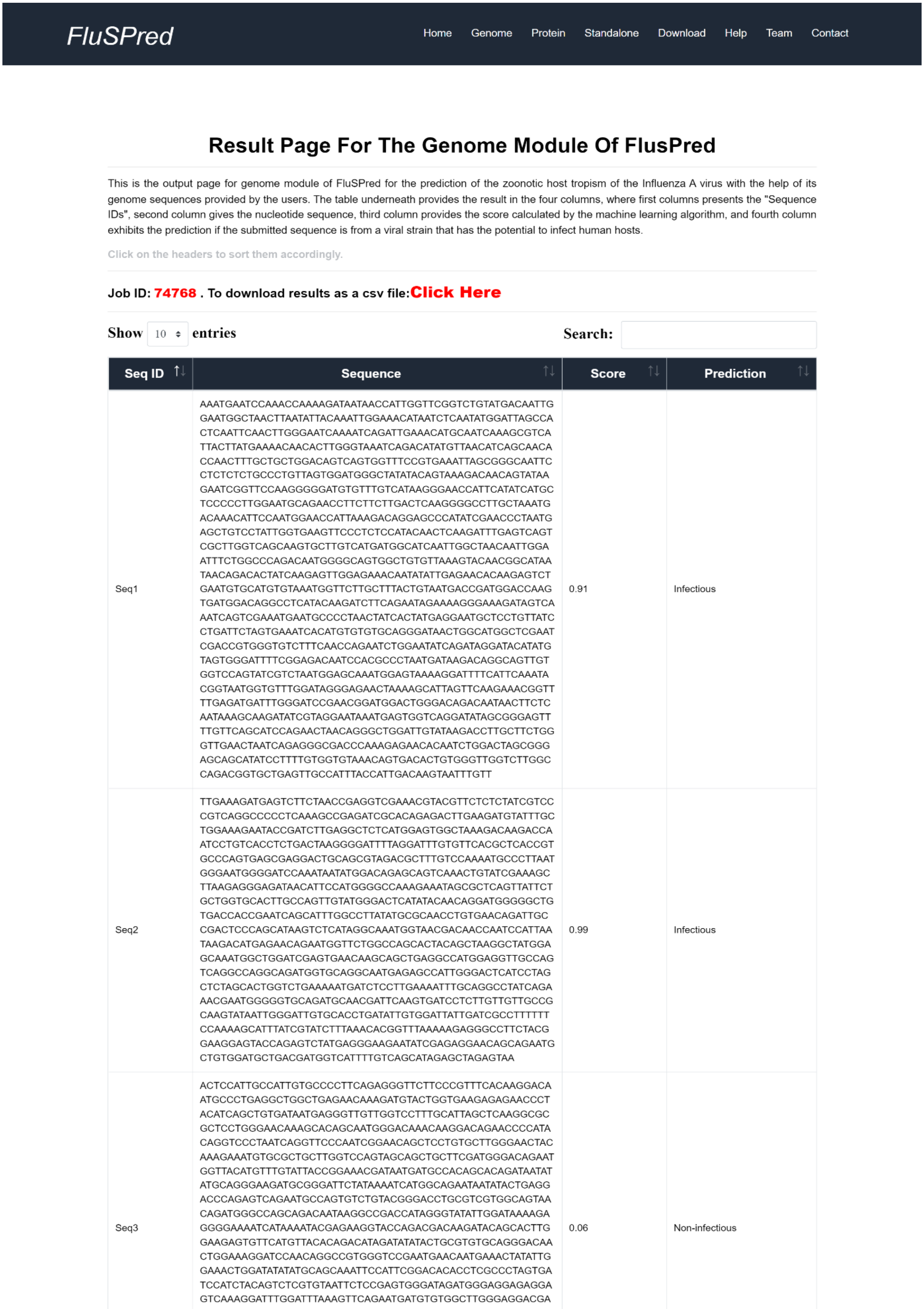
```
>Seq1
AAATGAATCCAAACCAAAAGATAATAACCATTGGTTCGGTCTGTATGACAATTGGAATGGCTAACTTAATATTACAAATTG
GAAACATAATCTCAATATGGATTAGCCACTCAATTCAACTTGGGAATCAAAATCAGATTGAAACATGCAATCAAAGCGTC
ATTACTTATGAAAACAACACTTGGGTAAATCAGACATATGTTAACATCAGCAACACCAACTTTGCTGCTGGACAGTCAGT
GGTTTCCGTGAAATTAGCGGGCAATTCCTCTCTCTGCCCTGTTAGTGGATGGGCTATATACAGTAAAGACAACAGTATA
AGAATCGGTTCCAAGGGGGATGTGTTTGTCATAAGGGAACCATTCATATCATGCTCCCCCTTGGAATGCAGAACCTTCT
TCTTGACTCAAGGGGCCTTGCTAAATGACAAACATTCCAATGGAACCATTAAAGACAGGAGCCCATATCGAACCCTAAT
```

OR Submit sequence file: Choose file No file chosen

Choose Probability Threshold: 0.5

Figure 4: Genome Module of FluSPred WebServer.

The result comprises the prediction of the human/non-human infectious strain, whether the particular strain would be capable of crossing the species barrier to infect humans or not. After the user clicks on the "Submit" button, FluSPred then runs the respective model and provides the result in the form of a table which can be downloaded in .csv format.



Result Page For The Genome Module Of FlusPred

This is the output page for genome module of FluSPred for the prediction of the zoonotic host tropism of the Influenza A virus with the help of its genome sequences provided by the users. The table underneath provides the result in the four columns, where first columns presents the "Sequence IDs", second column gives the nucleotide sequence, third column provides the score calculated by the machine learning algorithm, and fourth column exhibits the prediction if the submitted sequence is from a viral strain that has the potential to infect human hosts.

Click on the headers to sort them accordingly.

Job ID: 74768 . To download results as a csv file:Click Here

Show 10 entries Search:

Seq ID	Sequence	Score	Prediction
Seq1	AAATGAATCCAAACCAAAAGATAATAACCATTGGTTCGGTCTGTATGACAATTG GAATGGCTAACTTAATATTACAAATTGGAAACATAATCTCAATATGGATTAGCCA CTCAATTCAACTTGGGAATCAAAATCAGATTGAAACATGCAATCAAAGCGTCA TTACTTATGAAAACAACACTTGGGTAAATCAGACATATGTTAACATCAGCAACA CCAACTTTGCTGCTGGACAGTCAGTGGTTTCCGTGAAATTAGCGGGCAATTC CTCTCTCTGCCCTGTTAGTGGATGGGCTATATACAGTAAAGACAACAGTATAA GAATCGGTTCCAAGGGGGATGTGTTTGTCATAAGGGAACCATTCATATCATGC TCCCCCTTGGAATGCAGAACCTTCTTCTTGACTCAAGGGGCCTTGCTAAATG ACAAACATTCCAATGGAACCATTAAAGACAGGAGCCCATATCGAACCCTAATG AGCTGTCCTATTGGTGAAGTTCCCTCTCCATACAACTCAAGATTTGAGTCAGT CGCTTGGTCAGCAAGTGCTTGTCATGATGGCATCAATTGGCTAACAATTGGA ATTTCTGGCCCAGACAATGGGGCAGTGGCTGTGTTAAAGTACAACGGCATAA TAACAGACACTATCAAGAGTTGGAGAAACAATATATTGAGAACACAAGAGTCT GAATGTGCATGTGTAAATGGTTCTTGCTTTACTGTAATGACCGATGGACCAAG TGATGGACAGGCCTCATACAAGATCTTCAGAATAGAAAAGGGAAAGATAGTCA AATCAGTCGAAATGAATGCCCCTAACTATCACTATGAGGAATGCTCCTGTTATC CTGATTCTAGTGAAATCACATGTGTGTGCAGGGATAACTGGCATGGCTCGAAT CGACCGTGGGTGTCTTTCAACCAGAATCTGGAATATCAGATAGGATACATATG TAGTGGGATTTTCGGAGACAATCCACGCCCTAATGATAAGACAGGCAGTTGT GGTCCAGTATCGTCTAATGGAGCAAATGGAGTAAAAGGATTTTCATTCAAATA CGGTAATGGTGTTTGGATAGGGAGAACTAAAAGCATTAGTTCAAGAAACGGTT TTGAGATGATTTGGGATCCGAACGGATGGACTGGGACAGACAATAACTTCTC AATAAAGCAAGATATCGTAGGAATAAATGAGTGGTCAGGATATAGCGGGAGTT TTGTTCAGCATCCAGAACTAACAGGGCTGGATTGTATAAGACCTTGCTTCTGG GTTGAACTAATCAGAGGGCGACCCAAAGAGAACACAATCTGGACTAGCGGG AGCAGCATATCCTTTTGTGGTGTAAACAGTGACACTGTGGGTTGGTCTTGGC CAGACGGTGCTGAGTTGCCATTTACCATTGACAAGTAATTTGTT	0.91	Infectious
Seq2	TTGAAAGATGAGTCTTCTAACCGAGGTCGAAACGTACGTTCTCTCTATCGTCC CGTCAGGCCCCCTCAAAGCCGAGATCGCACAGAGACTTGAAGATGTATTTGC TGGAAAGAATACCGATCTTGAGGCTCTCATGGAGTGGCTAAAGACAAGACCA ATCCTGTCACCTCTGACTAAGGGGATTTTAGGATTTGTGTTCACGCTCACCGT GCCCAGTGAGCGAGGACTGCAGCGTAGACGCTTTGTCCAAAATGCCCTTAAT GGGAATGGGGATCCAAATAATATGGACAGAGCAGTCAAACTGTATCGAAAGC TTAAGAGGGAGATAACATTCCATGGGGCCAAAGAAATAGCGCTCAGTTATTCT GCTGGTGCACTTGCCAGTTGTATGGGACTCATATACAACAGGATGGGGGCTG TGACCACCGAATCAGCATTTGGCCTTATATGCGCAACCTGTGAACAGATTGC CGACTCCCAGCATAAGTCTCATAGGCAAATGGTAACGACAACCAATCCATTAA TAAGACATGAGAACAGAATGGTTCTGGCCAGCACTACAGCTAAGGCTATGGA GCAAATGGCTGGATCGAGTGAACAAGCAGCTGAGGCCATGGAGGTTGCCAG TCAGGCCAGGCAGATGGTGCAGGCAATGAGAGCCATTGGGACTCATCCTAG CTCTAGCACTGGTCTGAAAAATGATCTCCTTGAAAATTTGCAGGCCTATCAGA AACGAATGGGGGTGCAGATGCAACGATTCAAGTGATCCTCTTGTTGTTGCCG CAAGTATAATTGGGATTGTGCACCTGATATTGTGGATTATTGATCGCCTTTTTT CCAAAAGCATTTATCGTATCTTTAAACACGGTTTAAAAAGAGGGCCTTCTACG GAAGGAGTACCAGAGTCTATGAGGGAAGAATATCGAGAGGAACAGCAGAATG CTGTGGATGCTGACGATGGTCATTTTGTCAGCATAGAGCTAGAGTAA	0.99	Infectious
Seq3	ACTCCATTGCCATTGTGCCCCTTCAGAGGGTTCTTCCCGTTTCACAAGGACA ATGCCCTGAGGCTGGCTGAGAACAAAGATGTACTGGTGAAGAGAGAACCCT ACATCAGCTGTGATAATGAGGGTTGTTGGTCCTTTGCATTAGCTCAAGGCGC GCTCCTGGGAACAAAGCACAGCAATGGGACAAACAAGGACAGAACCCCATA CAGGTCCCTAATCAGGTTCCCAATCGGAACAGCTCCTGTGCTTGGGAACTAC AAAGAAATGTGCGCTGCTTGGTCCAGTAGCAGCTGCTTCGATGGGACAGAAT GGTTACATGTTTGTATTACCGGAAACGATAATGATGCCACAGCACAGATAATAT ATGCAGGGAAGATGCGGGATTCTATAAAATCATGGCAGAATAATATACTGAGG ACCCAGAGTCAGAATGCCAGTGTCTGTACGGGACCTGCGTCGTGGCAGTAA CAGATGGGCCAGCAGACAATAAGGCCGACCATAGGGTATATTGGATAAAAGA GGGGAAAATCATAAAATACGAGAAGGTACCAGACGACAAGATACAGCACTTG GAAGAGTGTTCATGTTACACAGACATAGATATATACTGCGTGTGCAGGGACAA CTGGAAAGGATCCAACAGGCCGTGGGTCCGAATGAACAATGAAACTATATTG GAAACTGGATATATATGCAGCAAATTCCATTCGGACACACCTCGCCCTAGTGA TCCATCTACAGTCTCGTGTAATTCTCCGAGTGGGATAGATGGGAGGAGAGGA GTCAAAGGATTTGGATTTAAAGTTCAGAATGATGTGTGGCTTGGGAGGACGA	0.06	Non-infectious

Figure 5: Result Page for Genome Module of FluSPred with example sequence

As a service to the scientific community, the webserver “FluSPred” was made open-source. This web-server can be used for further research related to Influenza A or public health and pandemic surveillance in the long run.

Chapter 3: Host Prediction of Rotavirus A

Workflow

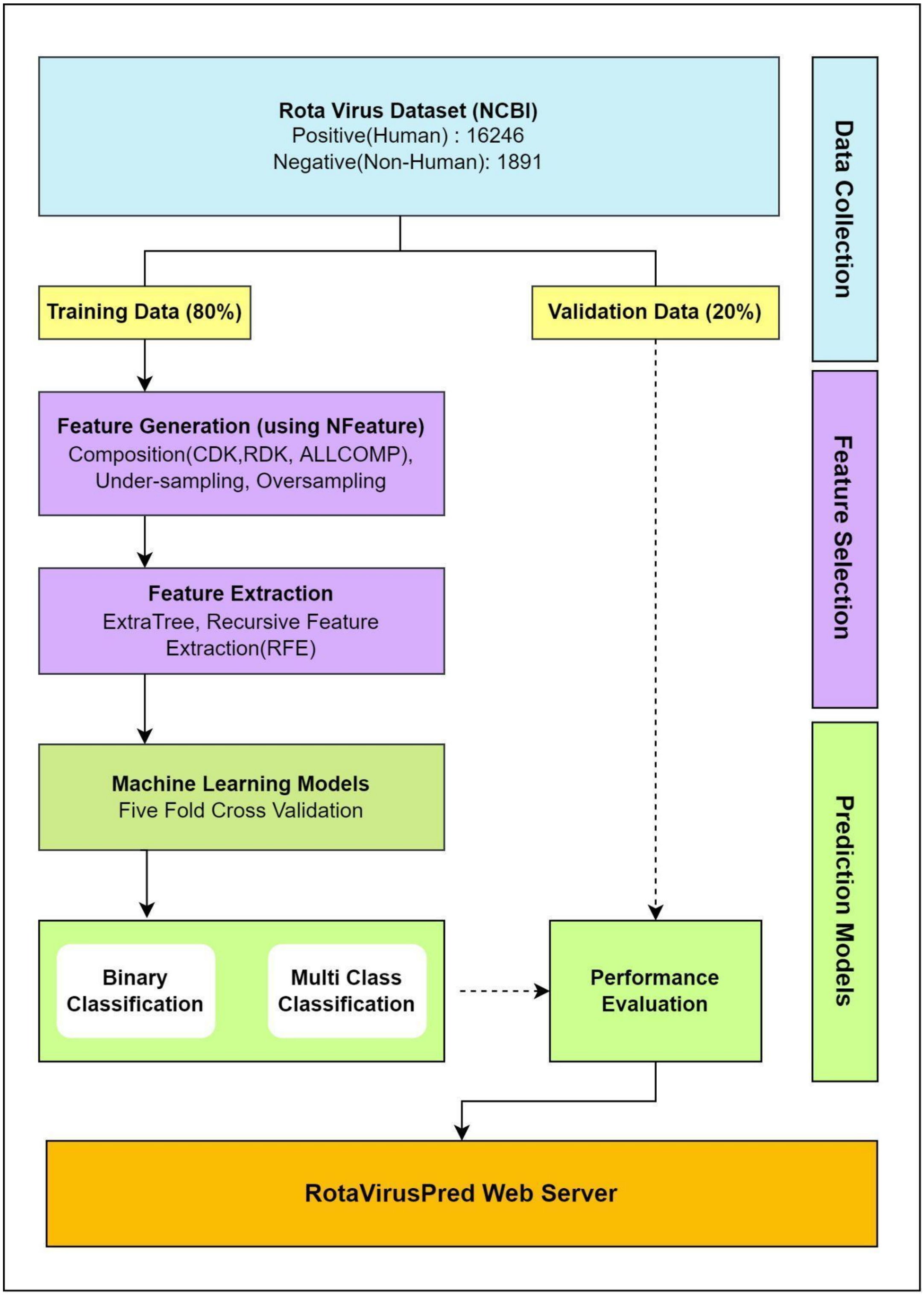


Figure 6: Workflow Diagram of RotaVirusPred Project from Data collection to Web Server

Material and Methods

Dataset Preparation

The dataset for genomic sequences of Rotavirus was obtained from Virus Pathogen Database and Analysis Resource (VIPR)[38]. 18137 total Rotavirus A genome sequences were taken which contained 16246 sequences derived from humans, which we have regarded as a “positive” dataset, and 1891 sequences derived from other animals which we regarded as a “negative” dataset. The dataset was then cleaned by removing duplicate sequences and unknown hosts. The positive and the negative datasets were labelled with values of 1 and 0 respectively. Here in this dataset, there was a high imbalance between the positive and negative sequences. Therefore we tried random oversampling the minority class data. The total number of sequences was divided into 80% training and 20% independent external validation data [25, 26].

Random Oversampling

Random Oversampling which randomly duplicates examples in the minority class, i.e. here it increased the number of negative data points [39]. Therefore the dataset has 16246 positive as well as negative sequences. This technique saves the possibility of data loss which is why there is sufficient data for training. However, it may increase the likelihood of overfitting to occur, since it makes exact copies of minority classes. It is also computer expensive as there is a substantial increase in the data[40].

Feature Generation

The features for the genome was computed using Nfeature where each nucleotide A T G C composition is taken out as per its K-Mers. CDK and All_Comp features were taken out from Nfeature, where both of them are composition based features[27]. CDK is a feature extraction method that calculates the composition of all 4 nucleotides. Here K could be 1, 2, or 3. All_comp is a feature extraction method which comprises of nucleotide based features such as CDK, RDK, nucleotide repeat index(NRI), distance distribution of nucleotides (DDN), entropy-based features(ES) and pseudo composition based features (PDNC). This generates a feature vector of 191 columns.

Feature Selection

Single-Feature Method

For the prediction of host of rotavirus based on its genome sequences Single feature method feature selection technique was conducted on the all_comp feature, which includes the CDK feature as well. Each feature from the all_comp features was selected one by one and for each column, the minimum and maximum of that column were calculated. Based on the minimum and maximum values the threshold was calculated for each column. Each column had four thresholds, i.e. min, min+max/4, min + max/2, (min+max)¾. Taking each threshold into account, then if a composition value was above the threshold, they were marked as one and if it was less than the threshold then it was marked zero. Based on this distribution we got values for each column called the prediction column. Apart from this, we had the actual column which had values of 1 for the human host and 0 for the non-human host for each sequence. We used both the columns and calculated the predictions for each of the columns. based on these scores we ranked the features of the all_comp feature extraction. Using this feature ranking technique we divided the top features in sets of top 5, top 10, top 15, top 20, and top 25 feature sets. For each ranked feature set we then applied machine learning models on them and obtained results for the same for comparison.

Machine Learning Models

In this study, many machine learning algorithms have been implemented for developing binary classification models that include K-Nearest Neighbour (KNN), Random Forest (RF), Logistic Regression(LR), Decision Tree (DT), Support Vector Machine (SVM), and XGBoost (XGB). Instance-based classification method KNN classifies based on the vote of the nearest neighbor data points [28]. It only stores the instances of the training variables, and the majority vote of the nearest neighbor of each data point determines classification. In comparison, RF is an ensemble-based method for classification. While training, the RF classifier fits numerous DTs to predict the response variable as an individual tree [29]. Averaging DT scores improves the prediction accuracy and control on the overfitting of the models. In contrast, DT is a non-parametric-based supervised learning algorithm that predicts the response variable by learning the decision rules from the data features[30]. XGB is an ensemble based method that implements the gradient boosted decision trees designed for speed and performance. It uses an iterative approach and provides a wrapper class to treat models like classifiers or regressors [31]. SVM is a supervised learning algorithm that finds a hyperplane in an N-dimensional space for the classification[32]. DT, KNN, XGB, SVM, LR, and RF-based models were generated for genome sequences using CDK and All_comp features. These classification techniques were implemented using python-library sci-kit learn [33].

Cross-Validation

Here, each model underwent 5-fold cross-validation. The datasets were divided into training and validation datasets. The training data was further divided into training and testing datasets during the 5 fold cross-validation process and the mean of the results for each fold of the cross-fold validation were noted down. In the 5-fold cross-validation process, the entire training data gets divided into 5 equivalent folds and then 4 folds were used for training and the 5th fold was used for testing. The whole procedure gets iterated five times where every fold gets a chance for being used as testing data. This is a standard procedure used in many studies[34-37].

Evaluation Parameters

In the current study, we have used the standard performance evaluation parameters such as accuracy, sensitivity, specificity, precision, recall, Matthews Correlation Coefficient (MCC), and Area Under the Curve (AUC) were computed. Whereas, accuracy, sensitivity, and specificity are threshold-dependent parameters and AUC is threshold independent parameter that is plotted by sensitivity vs 1-specificity. The parameters are computed by:

$$Sensitivity = \frac{T_P}{T_P+F_N} \qquad (vi)$$

$$Specificity = \frac{T_N}{T_N+F_P} \qquad (vii)$$

$$Accuracy = \frac{T_P+T_N}{T_P+T_N+F_P+F_N} \qquad (viii)$$

$$MCC = \frac{\left(T_P*T_N\right)-(F_P*F_N)}{\sqrt{(T_P+F_P)(T_P+F_N)(T_N+F_P)(T_N+F_N)}} \qquad (ix)$$

Where, T_P, T_N, F_P and F_N stand for true positive, true negative, false positive and false negative, respectively.

Results

Performance of Models based on RDK features

Using the RDK based features for both k=1 and 3, KNN showed the highest evaluation results with an accuracy of 98.6% and an AUC of 0.986 for k=3 in training dataset. For the validation dataset, this model showed 88.7% as accuracy and 0.918 AUC. Other machine learning algorithms have also been tried, whose results are shown in Table 11.

Table 11: Model performance using machine learning models based on RDK(K = 1, 3) features on training and validation dataset

		Training					Validation				
Model	**k-mer**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**
KNN	1	0.919	0.886	0.951	0.919	0.84	0.824	0.856	0.539	0.706	0.308
	3	0.986	0.973	1	0.986	0.973	0.887	0.817	0.95	0.918	0.778
RF	1	0.711	0.579	0.843	0.711	0.438	0.604	0.581	0.807	0.735	0.236
	3	0.747	0.683	0.811	0.747	0.499	0.709	0.629	0.78	0.798	0.415
SVM	1	0.702	0.632	0.772	0.702	0.408	0.644	0.633	0.745	0.692	0.232
	3	0.911	0.901	0.922	0.911	0.823	0.754	0.772	0.738	0.837	0.509
DT	1	0.951	0.917	0.985	0.951	0.905	0.842	0.879	0.512	0.705	0.322
	3	0.982	0.965	0.998	0.982	0.965	0.825	0.797	0.85	0.824	0.649
LR	1	0.655	0.716	0.595	0.655	0.313	0.702	0.718	0.56	0.697	0.183
	3	0.699	0.707	0.692	0.699	0.399	0.619	0.691	0.556	0.688	0.248
XGB	1	0.726	0.643	0.809	0.726	0.458	0.658	0.645	0.769	0.753	0.256
	3	0.914	0.906	0.922	0.914	0.829	0.833	0.82	0.845	0.889	0.665

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #LR: Logistic regression, XGB: XGBoost

Performance of Models based on CDK features

Here, Both DT and KNN show similar results for k=1 and k=3 respectively. KNN showed the highest evaluation results with an accuracy of 98.6% and an AUC of 0.986 for in training dataset. For the validation dataset, this model showed 90.8% as accuracy and 0.932 AUC. DT model has 96.9% accuracy and 0.978 AUC for training and 97.3 accuracy and 0.957 AUC in validation. The rest of the results are shown in Table 12.

Table 12: Model performance using machine learning models based on CDK(K = 1, 3) features on training and validation dataset

		Training					Validation				
Model	**k-mer**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**
KNN	1	0.986	0.973	1	0.986	0.974	0.959	0.92	0.999	0.974	0.921
	3	0.986	0.973	1	0.986	0.973	0.908	0.862	0.958	0.932	0.822
RF	1	0.748	0.671	0.824	0.747	0.502	0.7	0.61	0.792	0.764	0.409
	3	0.747	0.687	0.808	0.747	0.499	0.734	0.646	0.829	0.816	0.482
SVM	1	0.912	0.893	0.93	0.911	0.824	0.709	0.65	0.77	0.734	0.423
	3	0.914	0.901	0.928	0.914	0.829	0.819	0.786	0.854	0.888	0.64
DT	1	0.977	0.956	0.999	0.978	0.956	0.973	0.948	1	0.975	0.949
	3	0.979	0.961	0.996	0.978	0.959	0.825	0.842	0.807	0.824	0.65
LR	1	0.706	0.701	0.711	0.706	0.412	0.694	0.713	0.674	0.72	0.388
	3	0.698	0.709	0.687	0.698	0.397	0.659	0.592	0.73	0.686	0.325
XGB	1	0.916	0.907	0.924	0.916	0.832	0.747	0.699	0.795	0.834	0.497
	3	0.917	0.909	0.925	0.917	0.835	0.848	0.849	0.846	0.915	0.695

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #LR: Logistic regression, XGB: XGBoost

Performance of Models based on All_comp features

In this method, the best results have been shown by KNN with 98.6% accuracy as well as AUC in training data and 95.3% accuracy and 0.915 as AUC. The comprehensive results have been given in the table below(Table 13):

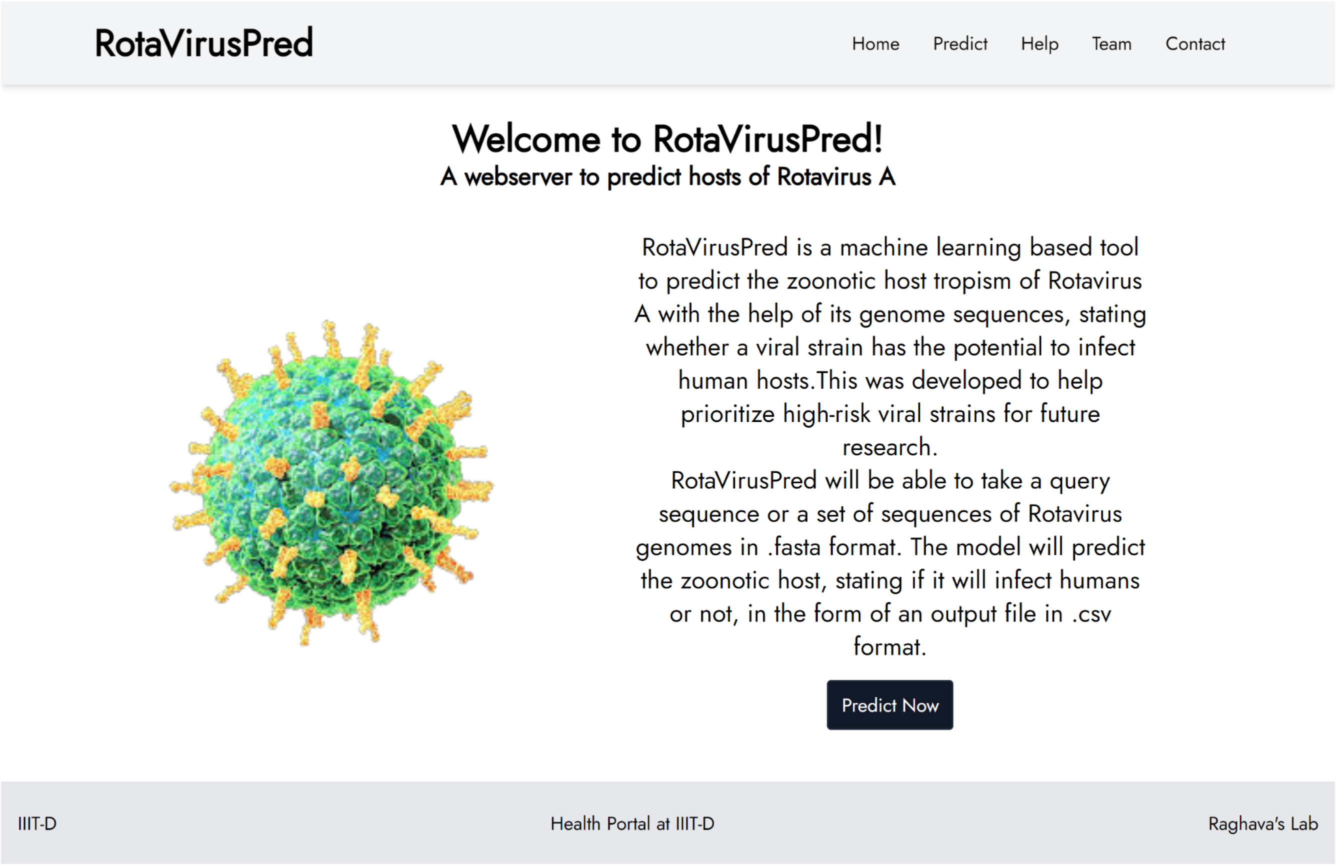
Table 13: Model performance using machine learning models based on ALLCOMP(K = 1, 3) features on training and validation dataset

		Training					Validation				
Model	**k-mer**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**	**Accuracy**	**Sensitivity**	**Specificity**	***AUC***	**MCC**
KNN	1	0.986	0.973	1	0.986	0.973	0.954	0.965	0.85	0.911	0.763
	3	0.986	0.972	1	0.986	0.973	0.953	0.965	0.855	0.915	0.774
RF	1	0.727	0.617	0.837	0.727	0.466	0.651	0.633	0.819	0.836	0.275
	3	0.748	0.673	0.821	0.747	0.501	0.68	0.662	0.832	0.837	0.312
SVM	1	0.678	0.672	0.683	0.678	0.356	0.686	0.686	0.691	0.729	0.236
	3	0.912	0.899	0.925	0.912	0.825	0.856	0.856	0.86	0.911	0.533
DT	1	0.978	0.957	0.999	0.978	0.958	0.939	0.972	0.641	0.807	0.646
	3	0.976	0.953	0.999	0.976	0.953	0.933	0.965	0.669	0.817	0.645
LR	1	0.674	0.696	0.653	0.674	0.35	0.708	0.717	0.625	0.72	0.219
	3	0.7	0.7	0.701	0.7	0.401	0.696	0.692	0.731	0.77	0.273
XGB	1	0.921	0.906	0.935	0.921	0.842	0.905	0.907	0.888	0.948	0.631
	3	0.918	0.907	0.929	0.918	0.836	0.898	0.899	0.894	0.955	0.63

#KNN: K-Nearest neighbor, #RF: Random F=forest, #SVM: Support vector machine, #DT: Decision tree, #LR: Logistic regression, XGB: XGBoost

Architechture of the Web-Server

We have made a webserver named “RotaVirusPred” (https://webs.iiitd.edu.in/raghava/rotavpred/) that the scientific community can use to predict human or non-human infectious strains of rotavirus A. The best model, i.e. KNN from All_comp features for k=3, was incorporated into the webserver. Web Server was built in two parts, Front End and Back End. HTML & CSS were used for styling and VanillaJS for form validation and sequence submission logic for the front end. The Backend used Python and PHP to run the prediction model and generate a result file. The website was made keeping the focus on mobile and laptop users, and this is why it is fully responsive and functional with all screen sizes.



Welcome to RotaVirusPred!

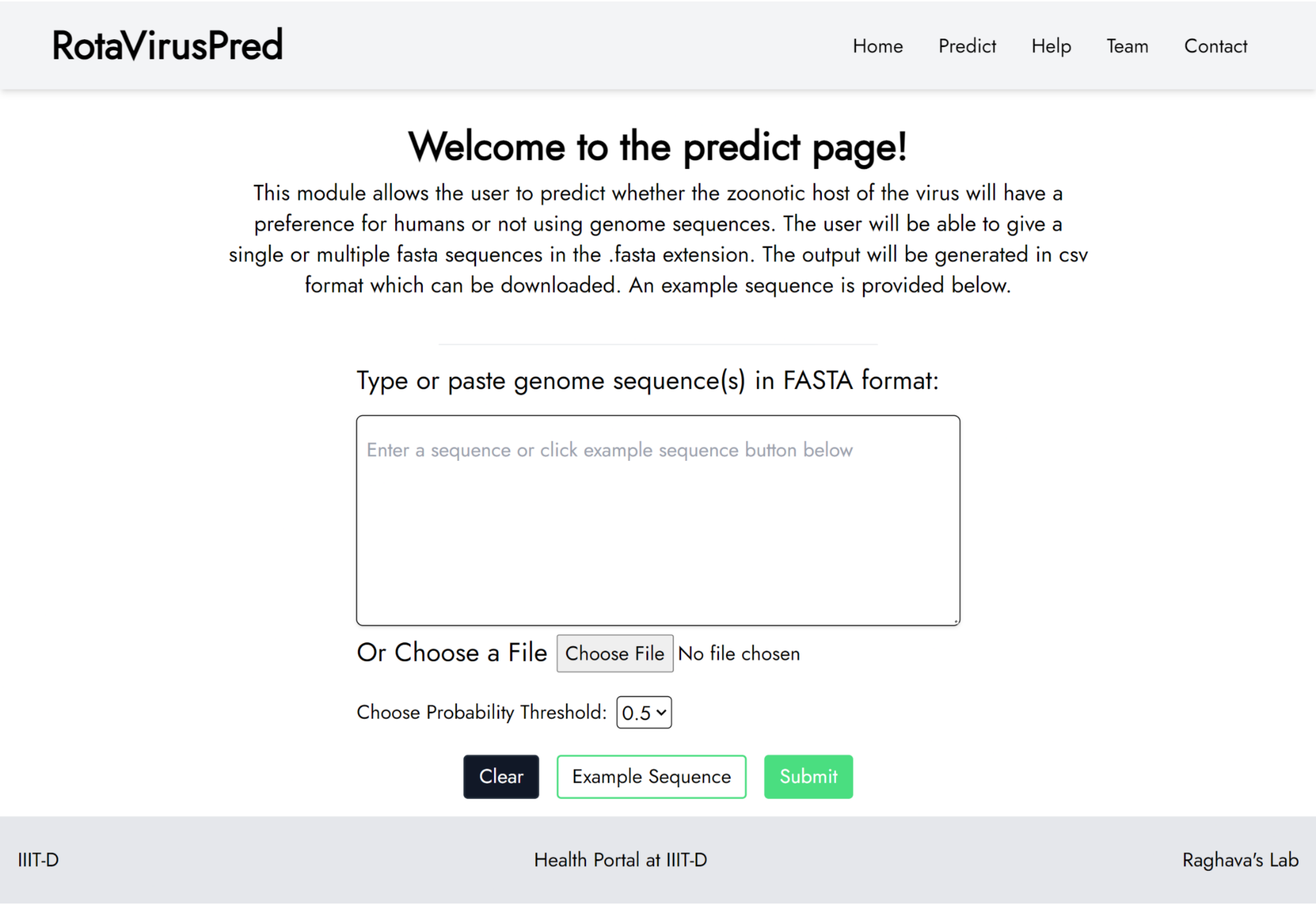
A webserver to predict hosts of Rotavirus A

RotaVirusPred is a machine learning based tool to predict the zoonotic host tropism of Rotavirus A with the help of its genome sequences, stating whether a viral strain has the potential to infect human hosts.This was developed to help prioritize high-risk viral strains for future research.

RotaVirusPred will be able to take a query sequence or a set of sequences of Rotavirus genomes in .fasta format. The model will predict the zoonotic host, stating if it will infect humans or not, in the form of an output file in .csv format.

Figure 7: RotaVirusPred Web Server Homepage

The Predict page has an input box that takes in a single or multiple sequences, rotavirus A genome, in .FASTA format.



Welcome to the predict page!

This module allows the user to predict whether the zoonotic host of the virus will have a preference for humans or not using genome sequences. The user will be able to give a single or multiple fasta sequences in the .fasta extension. The output will be generated in csv format which can be downloaded. An example sequence is provided below.

Type or paste genome sequence(s) in FASTA format:

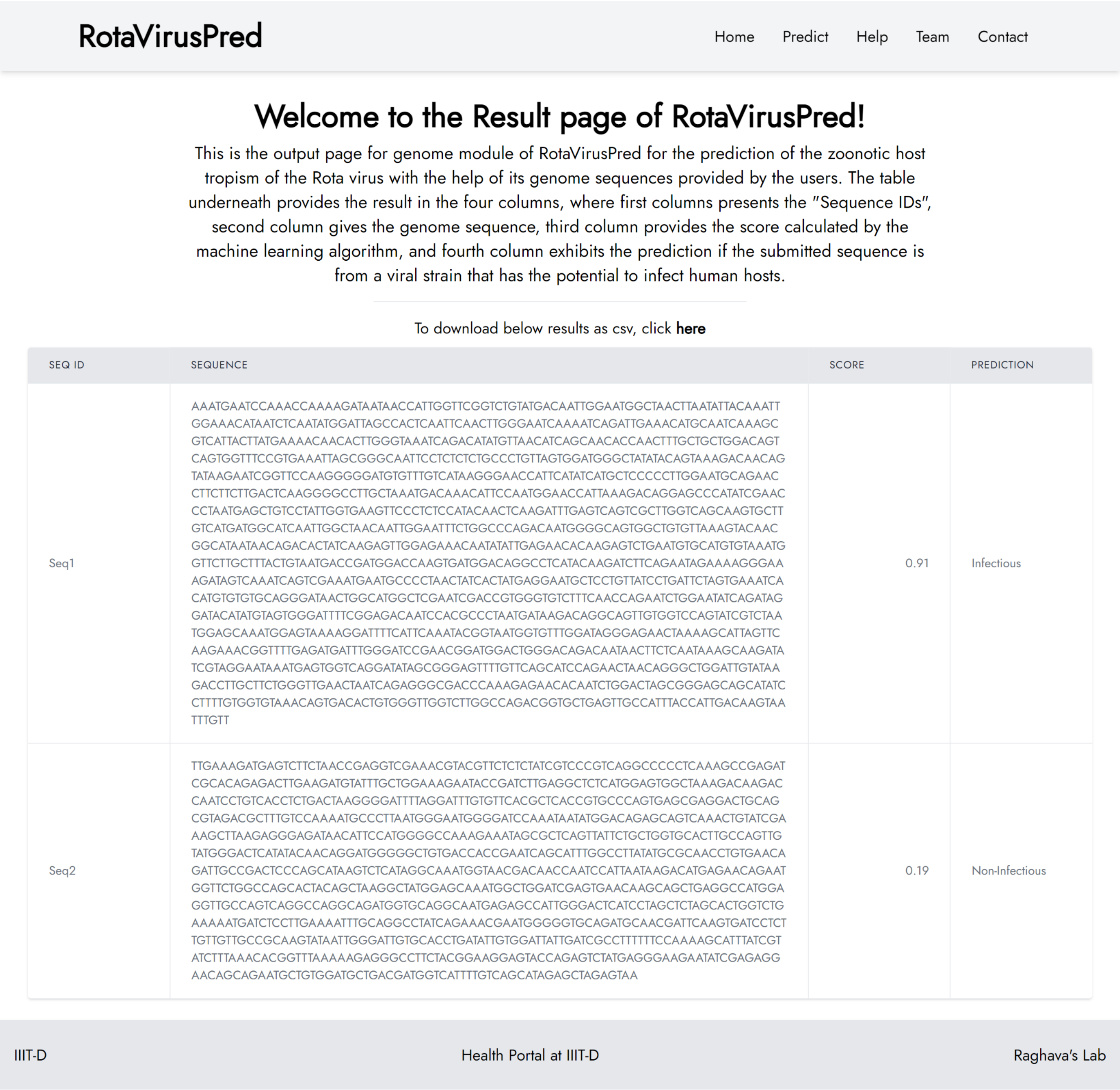
Enter a sequence or click example sequence button below

Or Choose a File Choose File No file chosen

Choose Probability Threshold: 0.5

Figure 8: RotaVirusPred Web Server Genome Prediction Module

After the user clicks on the "Submit" button, RotaVirusPred then runs the respective model and provides the result in the form of a table which can be downloaded in .csv format.



Welcome to the Result page of RotaVirusPred!

This is the output page for genome module of RotaVirusPred for the prediction of the zoonotic host tropism of the Rota virus with the help of its genome sequences provided by the users. The table underneath provides the result in the four columns, where first columns presents the "Sequence IDs", second column gives the genome sequence, third column provides the score calculated by the machine learning algorithm, and fourth column exhibits the prediction if the submitted sequence is from a viral strain that has the potential to infect human hosts.

To download below results as csv, click **here**

SEQ ID	SEQUENCE	SCORE	PREDICTION
Seq1	AAATGAATCCAAACCAAAAGATAATAACCATTGGTTCGGTCTGTATGACAATTGGAATGGCTAACTTAATATTACAAATTGGAAACATAATCTCAATATGGATTAGCCACTCAATTCAACTTGGGAATCAAAATCAGATTGAAACATGCAATCAAAGCGTCATTACTTATGAAAACAACACTTGGGTAAATCAGACATATGTTAACATCAGCAACACCAACTTTGCTGCTGGACAGTCAGTGGTTTCCGTGAAATTAGCGGGCAATTCCTCTCTCTGCCCTGTTAGTGGATGGGCTATATACAGTAAAGACAACAGTATAAGAATCGGTTCCAAGGGGGATGTGTTTGTCATAAGGGAACCATTCATATCATGCTCCCCCTTGGAATGCAGAACCTTCTTCTTGACTCAAGGGGCCTTGCTAAATGACAAACATTCCAATGGAACCATTAAAGACAGGAGCCCATATCGAACCCTAATGAGCTGTCCTATTGGTGAAGTTCCCTCTCCATACAACTCAAGATTTGAGTCAGTCGCTTGGTCAGCAAGTGCTTGTCATGATGGCATCAATTGGCTAACAATTGGAATTTCTGGCCCAGACAATGGGGCAGTGGCTGTGTTAAAGTACAACGGCATAATAACAGACACTATCAAGAGTTGGAGAAACAATATATTGAGAACACAAGAGTCTGAATGTGCATGTGTAAATGGTTCTTGCTTTACTGTAATGACCGATGGACCAAGTGATGGACAGGCCTCATACAAGATCTTCAGAATAGAAAAGGGAAAGATAGTCAAATCAGTCGAAATGAATGCCCCTAACTATCACTATGAGGAATGCTCCTGTTATCCTGATTCTAGTGAAATCACATGTGTGTGCAGGGATAACTGGCATGGCTCGAATCGACCGTGGGTGTCTTTCAACCAGAATCTGGAATATCAGATAGGATACATATGTAGTGGGATTTTCGGAGACAATCCACGCCCTAATGATAAGACAGGCAGTTGTGGTCCAGTATCGTCTAATGGAGCAAATGGAGTAAAAGGATTTTCATTCAAATACGGTAATGGTGTTTGGATAGGGAGAACTAAAAGCATTAGTTCAAGAAACGGTTTTGAGATGATTTGGGATCCGAACGGATGGACTGGGACAGACAATAACTTCTCAATAAAGCAAGATATCGTAGGAATAAATGAGTGGTCAGGATATAGCGGGAGTTTTGTTCAGCATCCAGAACTAACAGGGCTGGATTGTATAAGACCTTGCTTCTGGGTTGAACTAATCAGAGGGCGACCCAAAGAGAACACAATCTGGACTAGCGGGAGCAGCATATCCTTTTGTGGTGTAAACAGTGACACTGTGGGTTGGTCTTGGCCAGACGGTGCTGAGTTGCCATTTACCATTGACAAGTAATTTGTT	0.91	Infectious
Seq2	TTGAAAGATGAGTCTTCTAACCGAGGTCGAAACGTACGTTCTCTCTATCGTCCCGTCAGGCCCCCTCAAAGCCGAGATCGCACAGAGACTTGAAGATGTATTTGCTGGAAAGAATACCGATCTTGAGGCTCTCATGGAGTGGCTAAAGACAAGACCAATCCTGTCACCTCTGACTAAGGGGATTTTAGGATTTGTGTTCACGCTCACCGTGCCCAGTGAGCGAGGACTGCAGCGTAGACGCTTTGTCCAAAATGCCCTTAATGGGAATGGGGATCCAAATAATATGGACAGAGCAGTCAAACTGTATCGAAAGCTTAAGAGGGAGATAACATTCCATGGGGCCAAAGAAATAGCGCTCAGTTATTCTGCTGGTGCACTTGCCAGTTGTATGGGACTCATATACAACAGGATGGGGGCTGTGACCACCGAATCAGCATTTGGCCTTATATGCGCAACCTGTGAACAGATTGCCGACTCCCAGCATAAGTCTCATAGGCAAATGGTAACGACAACCAATCCATTAATAAGACATGAGAACAGAATGGTTCTGGCCAGCACTACAGCTAAGGCTATGGAGCAAATGGCTGGATCGAGTGAACAAGCAGCTGAGGCCATGGAGGTTGCCAGTCAGGCCAGGCAGATGGTGCAGGCAATGAGAGCCATTGGGACTCATCCTAGCTCTAGCACTGGTCTGAAAAATGATCTCCTTGAAAATTTGCAGGCCTATCAGAAACGAATGGGGGTGCAGATGCAACGATTCAAGTGATCCTCTTGTTGTTGCCGCAAGTATAATTGGGATTGTGCACCTGATATTGTGGATTATTGATCGCCTTTTTTCCAAAAGCATTTATCGTATCTTTAAACACGGTTTAAAAAGAGGGCCTTCTACGGAAGGAGTACCAGAGTCTATGAGGGAAGAATATCGAGAGGAACAGCAGAATGCTGTGGATGCTGACGATGGTCATTTTGTCAGCATAGAGCTAGAGTAA	0.19	Non-Infectious

IIIT-D

Health Portal at IIIT-D

Raghava's Lab

Figure 9: RotaVirusPred Web Server Result page with Example Sequence

Discussion

Approximately 1 billion cases of zoonotic diseases are reported annually, including novel infectious agents [4]. With the increase in human migration and global reachability, the frequency is increasing in humans[41, 42]. The field of computational biology has seen quite an exceptional advancement in recent years. With this technology, it is now possible to have early disease warning systems.

Zoonotic diseases, including novel infectious agents, give rise to approximately 1 billion cases annually. [4]. With increasing human activity or interference, the frequency of zoonotic diseases in humans is increasing [41, 42].With the advancements in computational biology, it is now possible to have forecasting tools and early warning systems in disease outbreak analysis. Identifying strains that have this capability of causing a disease emergence and if it can encourage host permissiveness of zoonotic infections is the need of the hour[43].

We have made a systematic attempt to develop computational genome based models for the prediction of zoonotic hosts of the Influenza A and Rotavirus A. The sequence datasets were obtained from VIPR and VIDHOP, on which, compositional based feature extraction was conducted. The compositional feature based models showed high results, which indicates that simple composition based techniques can identify the important features and help in the prediction, as validated by the compositional analysis, which shows the difference in composition between the positive and the negative datasets. Random Forest models of CDK features for Influenza A and KNN model for All_comp feature for Rotavirus A for k = 3, were chosen as the best models.

Scientists have made several attempts to make computational tools to predict zoonotic events of different pathogens. Most of the available tools are complex and do not have a web-server which the community can use. Further, our models require genome sequences of the virus from the respective hosts. It does not require the sequences of the hosts, which can be cumbersome since their limited availability [44-46]. Zhang et al.[15] and Galiez et al,[17] combined species and genera to higher taxonomic groups whereas our approach takes account of the host species for each strains of the virus. *Mock et al* [24] performed deep learning, which is computer expensive, on multiclass classification of genomic data that was imbalanced, having preference on certain species over the others. They have achieved an average accuracy of 97.46 and an AUC of 0.94 on the influenza A dataset. On the other hand, we have performed binary classification with a focus on human and non-human hosts, achieving a higher prediction accuracy and AUC of 98% by using simple composition-based features, which are not computer expensive and time consuming. Li and Sun [16] used SVM, alignment-based and without alignment based methods to predict the host of influenza A genomic data. Their dataset was small

(1200 sequences and 6 hosts), and the average accuracy was 84%, 85.67%, and 87% for alignment based, SVM and alignment free methods respectively[16, 24]. Our models included a wide range of data, with 308632 sequences pertaining to 34 hosts as well as achieved a much higher accuracy.

We have provided two webservers, i.e., FluSPred and RotaVPred. Both of them are user-friendly. FluSPred is the first web server that provides agenome-based prediction model at a single place for the influenza A virus. RotaVPred, on the other hand is the only web server that exists for genome-based host transmission prediction for RotaVirus A, as of now. These servers use a machine-learning-based algorithm trained by compositional features such as (CDK, RDK, ALLCOMP) from Nfeature [27]
The compositional-based feature extraction is much faster and less resource-hungry than the rest of the methods discussed, yet highly accurate, as shown in the results. The user would only have to provide the sequence(s) if they wanted to use the webserver, which makes it very different from other existing methods. Based on the sequence, the server will tell if the particular sequence is transmissible to humans or not and also gives a probability score which the user can modify based threshold given while submitting the sequence. The datasets on which these servers are trained, covers a wide range of disease-causing viral genome sequences. The primary purpose of this web server is to serve the scientific community for predicting the zoonotic risk of the virus as a part of the early disease warning system.

Conclusion

Despite exceptional advances in science, healthcare, and medicine in the past century, we are still surrounded by bacteria, viruses, protoza, and fungi that are constantly evolving. Our research should majorly revolve around the root reason for cross-species transmission and finding new outbreaks before it starts spreading. This is why epidemiological disease surveillance systems are the need of the hour. An effective disease surveillance system can detect a significant outbreak early and save millions of lives and have less impact on the healthcare resources. Providing two open source website, one on Influenza A (FluSPred) and another on Rotavirus A (RotaVPred), we tried to play a small role as researchers in aiding to disease surveillance and early warning systems with a hope that this would help control, mittigate and most importantly prevent such events from occurring.

Future Objectives

FluSpred can classify between human and non-humans transmission, and reservoir sequence belongs to. There is scope of research that can be conducted on the same species' transmission and how the virus mutates on a single nucleotide level.

Furthermore, optimization in the existing algorithm of FluSPred and RotaVirusPred would help making the models more efficient and robust. The response time of the webserver from the moment the query is submitted and the result generated is high and can be reduced further.

References

1. Kruse, H., A.M. kirkemo, and K. Handeland, *Wildlife as source of zoonotic infections.* Emerg Infect Dis, 2004. **10**(12): p. 2067-72.
2. Levitt, A.M., A.S. Khan, and J.M. Hughes, *Chapter 4 - Emerging and re-emerging pathogens and diseases*, in *Infectious Diseases (Third Edition)*, J. Cohen, S.M. Opal, and W.G. Powderly, Editors. 2010, Mosby: London. p. 56-69.
3. Heeney, J.L., *Zoonotic viral diseases and the frontier of early diagnosis, control and prevention.* J Intern Med, 2006. **260**(5): p. 399-408.
4. Taylor, L.H., S.M. Latham, and M.E. Woolhouse, *Risk factors for human disease emergence.* Philos Trans R Soc Lond B Biol Sci, 2001. **356**(1411): p. 983-9.
5. Chomel, B.B., *Zoonoses.* Encyclopedia of Microbiology, 2009: p. 820-829.
6. Kilpatrick, A.M. and S.E. Randolph, *Drivers, dynamics, and control of emerging vector-borne zoonotic diseases.* Lancet, 2012. **380**(9857): p. 1946-55.
7. Nelson, M.I. and A.L. Vincent, *Reverse zoonosis of influenza to swine: new perspectives on the human-animal interface.* Trends Microbiol, 2015. **23**(3): p. 142-53.
8. Dhama, K., et al., *SARS-CoV-2 jumping the species barrier: Zoonotic lessons from SARS, MERS and recent advances to combat this pandemic virus.* Travel Med Infect Dis, 2020. **37**: p. 101830.
9. Kong, F., et al., *Porcine Deltacoronaviruses: Origin, Evolution, Cross-Species Transmission and Zoonotic Potential.* Pathogens, 2022. **11**(1).
10. Etienne, K.A., et al., *Whole-Genome Sequencing to Determine Origin of Multinational Outbreak of Sarocladium kiliense Bloodstream Infections.* Emerg Infect Dis, 2016. **22**(3): p. 476-81.
11. Whitworth, J., *COVID-19: a fast evolving pandemic.* Trans R Soc Trop Med Hyg, 2020. **114**(4): p. 241-248.
12. Peiris, J.S., M.D. de Jong, and Y. Guan, *Avian influenza virus (H5N1): a threat to human health.* Clin Microbiol Rev, 2007. **20**(2): p. 243-67.
13. Muller, H. and R. Johne, *Rotaviruses: diversity and zoonotic potential--a brief review.* Berl Munch Tierarztl Wochenschr, 2007. **120**(3-4): p. 108-12.
14. Brown, J.D., D.E. Stallknecht, and D.E. Swayne, *Experimental infection of swans and geese with highly pathogenic avian influenza virus (H5N1) of Asian lineage.* Emerg Infect Dis, 2008. **14**(1): p. 136-42.
15. Zhang, M., et al., *Prediction of virus-host infectious association by supervised learning methods.* BMC Bioinformatics, 2017. **18**(Suppl 3): p. 60.
16. Li, H. and F. Sun, *Comparative studies of alignment, alignment-free and SVM based approaches for predicting the hosts of viruses based on viral sequences.* Sci Rep, 2018. **8**(1): p. 10032.

17. Galiez, C., et al., *WIsH: who is the host? Predicting prokaryotic hosts from metagenomic phage contigs.* Bioinformatics, 2017. **33**(19): p. 3113-3114.
18. Ahlgren, N.A., et al., *Alignment-free d_2^* oligonucleotide frequency dissimilarity measure improves prediction of hosts from metagenomically-derived viral sequences.* Nucleic Acids Res, 2017. **45**(1): p. 39-53.
19. Edwards, R.A., et al., *Computational approaches to predict bacteriophage-host relationships.* FEMS Microbiol Rev, 2016. **40**(2): p. 258-72.
20. Wang, W., et al., *A network-based integrated framework for predicting virus-prokaryote interactions.* NAR Genom Bioinform, 2020. **2**(2): p. lqaa044.
21. Allen, T., et al., *Global hotspots and correlates of emerging zoonotic diseases.* Nat Commun, 2017. **8**(1): p. 1124.
22. Walker, J.W., et al., *Transmissibility of emerging viral zoonoses.* PLoS One, 2018. **13**(11): p. e0206926.
23. Agany, D.D.M., J.E. Pietri, and E.Z. Gnimpieba, *Assessment of vector-host-pathogen relationships using data mining and machine learning.* Comput Struct Biotechnol J, 2020. **18**: p. 1704-1721.
24. Mock, F., et al., *VIDHOP, viral host prediction with deep learning.* Bioinformatics, 2021. **37**(3): p. 318-325.
25. Dhall, A., et al., *Computing Skin Cutaneous Melanoma Outcome From the HLA-Alleles and Clinical Characteristics.* Front Genet, 2020. **11**: p. 221.
26. Gupta, S., et al., *In silico approach for predicting toxicity of peptides and proteins.* PLoS One, 2013. **8**(9): p. e73957.
27. Mathur, M., et al., *Nfeature: A platform for computing features of nucleotide sequences.* bioRxiv, 2021: p. 2021.12.14.472723.
28. Zhang, Z., *Introduction to machine learning: k-nearest neighbors.* Ann Transl Med, 2016. **4**(11): p. 218.
29. Rigatti, S.J., *Random Forest.* J Insur Med, 2017. **47**(1): p. 31-39.
30. Kingsford, C. and S.L. Salzberg, *What are decision trees?* Nat Biotechnol, 2008. **26**(9): p. 1011-3.
31. Griffis, J.C., J.B. Allendorfer, and J.P. Szaflarski, *Voxel-based Gaussian naive Bayes classification of ischemic stroke lesions in individual T1-weighted MRI scans.* J Neurosci Methods, 2016. **257**: p. 97-108.
32. Noble, W.S., *What is a support vector machine?* Nat Biotechnol, 2006. **24**(12): p. 1565-7.
33. Bac, J., et al., *Scikit-Dimension: A Python Package for Intrinsic Dimension Estimation.* Entropy (Basel), 2021. **23**(10).
34. Agrawal, P., et al., *In Silico Approach for Prediction of Antifungal Peptides.* Front Microbiol, 2018. **9**: p. 323.
35. Nagpal, G., et al., *Computer-aided prediction of antigen presenting cell modulators for designing peptide-based vaccine adjuvants.* J Transl Med, 2018. **16**(1): p. 181.
36. Qureshi, A., N. Thakur, and M. Kumar, *VIRsiRNApred: a web server for predicting inhibition efficacy of siRNAs targeting human viruses.* J Transl Med, 2013. **11**: p. 305.
37. Patiyal, S., et al., *NAGbinder: An approach for identifying N-acetylglucosamine interacting residues of a protein from its primary sequence.* Protein Sci, 2020. **29**(1): p. 201-210.
38. Pickett, B.E., et al., *ViPR: an open bioinformatics database and analysis resource for virology research.* Nucleic Acids Res, 2012. **40**(Database issue): p. D593-8.
39. Shamsudin, H., et al. *Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset*. in *2020 IEEE 16th International Conference on Control & Automation (ICCA)*. 2020.
40. Guo, X., et al. *On the Class Imbalance Problem*. in *2008 Fourth International Conference on Natural Computation*. 2008.
41. McArthur, D.B., *Emerging Infectious Diseases.* Nurs Clin North Am, 2019. **54**(2): p. 297-311.
42. Jones, K.E., et al., *Global trends in emerging infectious diseases.* Nature, 2008. **451**(7181): p. 990-3.

43. Han, B.A., et al., *Rodent reservoirs of future zoonotic diseases.* Proc Natl Acad Sci U S A, 2015. **112**(22): p. 7039-44.
44. Dilcher, M., et al., *Genetic characterization of Tribec virus and Kemerovo virus, two tick-transmitted human-pathogenic Orbiviruses.* Virology, 2012. **423**(1): p. 68-76.
45. Teeling, E.C., et al., *Bat Biology, Genomes, and the Bat1K Project: To Generate Chromosome-Level Genomes for All Living Bat Species.* Annu Rev Anim Biosci, 2018. **6**: p. 23-46.
46. Pagel Van Zee, J., et al., *Tick genomics: the Ixodes genome project and beyond.* Int J Parasitol, 2007. **37**(12): p. 1297-305.