MEDINFO 2021: One World, One Health – Global Partnership for Digital Innovation
P. Otero et al. (Eds.)
© 2022 International Medical Informatics Association (IMIA) and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/SHTI220167

Identifying New COVID-19 Variants from Spike Proteins Using Novelty Detection

Sayantani Basu[a], Roy H. Campbell[a]

[a] *Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, United States*

Abstract

The COVID-19 pandemic has caused millions of infections and deaths worldwide in an ongoing pandemic. With the passage of time, several variants of this virus have surfaced. Machine learning methods and algorithms have been very useful in understanding the virus and its implications so far. In this paper, we have studied a set of novelty detection algorithms and applied it to the problem of detecting COVID-19 variants. Our results show accuracies of 79.64% and 82.43% on the B.1.1.7 and B.1.351 variants respectively on ProtVec unaligned COVID-19 spike protein sequences using One Class SVM with fine-tuned parameters. We believe that a system for automated and timely detection of variants will help countries formulate mitigation measures and study remedies in terms of medicines and vaccines that can protect against the new variants.

Keywords:

Coronavirus [B04.820.578.500.540.150], Machine Learning [L01.224.050.375.530], Proteins [D12.776]

Introduction

The Corona Virus Disease (COVID-19), caused by the SARS-CoV-2 virus, has resulted in a global pandemic with a total of 144 million infections and 3 million deaths [1]. In an effort to curb the spread of the virus, countries worldwide have been practicing mitigation measures in the form of social distancing, wearing face coverings, frequent sanitizing, lockdowns, and phased reopenings. The spread of the virus has prompted several researchers to study the nature of the virus. There have been several proposed medications and remedies for COVID-19. More recently, several COVID-19 vaccines have been formulated in an effort to help combat the virus. However, several new variants of the COVID-19 virus have emerged over time. As a result, it is important to study these variants to help plan appropriate interventions or mitigation measures. In this paper, we formulate the identification of new COVID-19 variants as a novelty detection problem and perform a comparison of several novelty detection algorithms to determine a suitable method for automatically detecting COVID-19 variants. We perform case studies on the B.1.1.7 variant and the B.1.351 variant and show that the One Class SVM model performs best at detecting COVID-19 variants. Our model is useful in the current scenario for the purpose of detecting new variants.

Machine learning methods have been very useful in COVID-19 so far. There have been methods applied to X-ray image classification [2, 3], NLP based methods on text [4, 5], epidemiological data [6, 7], and biological sequence data [8, 9]. The challenges with studying COVID-19 data are as follows: (i) the virus is relatively new (records are available from December 2019) compared to other viruses like HIV (Human Immunodeficiency Virus) [10] or influenza [11] (ii) there are new records added everyday as this is an ongoing pandemic.

In this paper, we study a set of novelty detection methods on COVID-19 spike proteins and evaluate their performance based on identification of new variants. We discuss our methods and provide our results on the B.1.1.7 and B. 1. 351 variants.

Methods

We now discuss the methods used for our study. For pre-processing and converting the spike protein sequences to a set of features, we used two approaches, namely ProtVec [12] and 3-mers. For our study, the main objective was to identify new COVID-19 variants using novelty detection methods. These methods are inspired by the biological process of “novelty detection” whereby an organism identifies a process as “new” if it has not been encountered before.

Pre-processing and Feature Construction

ProtVec

ProtVec [12] is a set of 100-dimensional representations of various combinations of three residues of proteins, along with a vector representation for ‘unknown’. It is possible to use ProtVec for both aligned and unaligned protein sequences. In the present work, we construct the 100-dimensional representation for a protein sequence by adding the ProtVec vectors of overlapping residues of consecutive protein residues considered three at a time. For every group of three protein residues containing the ‘gaps’ in the aligned protein sequences, we add the unknown token. As part of our study, we have also reduced the number of features using principal components analysis (PCA) retaining at least 95% variance in the data and report results on both the original vectors as well as the data after PCA has been applied.

3-mers

A biological sequence can be decomposed into a set of substrings of length ‘k’. These substrings are known as k-mers. In this context, we implemented all our experiments using 3-mers in order to compare the results with the ProtVec approach. We collect all the frequencies of the 3-mer substring and arrange them in the form of a vector representation for each of the protein sequences. This forms our set of features for the 3-mer approach. However, it is to be noted at this point that the 3-mer approach generates a considerably large number of 3110 features that have to be then provided as input to the novelty detection algorithms. This was computationally intensive, and therefore, to simulate real-world situations where results are needed in a timely manner, we reduced the number of features using principal components analysis (PCA) to 268 dimensions and only report results on the reduced number of dimensions that capture at least 95% variance of the data.

Novelty Detection Algorithms

Elliptic Envelope

Elliptic Envelope is a method proposed by Rousseeuw and Driessen [13] that can be used for novelty detection. It is primarily used for detecting outliers in a setting where the data is assumed to be Gaussian.

Isolation Forest

Isolation Forest is a method proposed by Liu et al. [14, 15] that can be applied to novelty detection. It works by randomly splitting and partioning features based on the maximum and minimum value and building trees out of the splits. The forest of such trees containing shorter paths serves to 'isolate' the anomalies.

Local Outlier Factor

Local Outlier Factor [16] is a method that can be applied to novelty detection. It works by identifying the deviation in density of an observation with respect to its neighbors. The densities are compared to identify outliers with densities lower than their neighbors. We set 'novelty' parameter to 'True' in sklearn while using this approach.

One Class SVM

One Class SVM [17] is used to find anomalies based on support. It can be used with high-dimensional distributions and works with SVM (Support Vector Machine) as its basic algorithm.

Dimensionality Reduction

Principal Components Analysis

Principal Components Analysis (PCA) is a common dimensionality reduction technique. We use the sklearn implementation of probabilistic PCA [18] to retain at least 95% variance in the data. For the ProtVec approach, PCA reduced the dimensions from 100 to 1 and for the 3-mer approach, PCA reduced the dimensions from 3110 to 268.

Sequence Alignment

MAFFT

In order to carry out multiple sequence alignment on the protein sequences, we use MAFFT v7.475 [19] with default parameter settings. The input is passed as the set of unaligned protein sequences in FASTA format, and the output is the set of aligned protein sequences with gaps ('-').

Datasets

Spike Proteins

Spike proteins were obtained from the GISAID EpiCoV$^{\text{TM}}$ database in FASTA format that contains the sequence header and sequence data in unaligned form. For our latest analysis, we downloaded the spike proteins till April 7.

ProtVec protein vectors

The ProtVec pre-trained 3-grams were downloaded from Harvard Dataverse [20]. This dataset contains the 100-dimensional vector representation for each 3-gram. We use our own code to obtain the vector representation for each protein sequence (both in the aligned as well as unaligned cases) as previously discussed in the 'pre-processing and feature construction' subsection.

Sequence metadata

In order to validate and perform a comparison of methods used for novelty detection, sequence metadata is needed, especially the date of collection and the variant type. In a real-life setting, such a method can be deployed without the need of labels. However, the metadata is considered in order to quantify the metrics for our present study. The sequence metadata was collected from Nextstrain [21, 22]. The GISAID unique numbers were used to map the sequence metadata to their corresponding spike protein sequences.

Implementation

All pre-processing and programs were implemented using Python3. We used pandas [23] for pre-processing datasets and scikit-learn [24] for the dimensionality reduction and novelty detection algorithms.

Results

We now present the results of our study on COVID-19 variant detection. As previously discussed in the Methods section, we have used ProtVec and 3-mers respectively on the unaligned and aligned COVID-19 spike protein sequences. This provides us with the vector representations to which we then apply the variant detection methods. Table 1 shows the time taken by various pre-processing methods. We do not include the processes of aligning the sequences and generating labels since the same amount of time was taken for these processes. The time in seconds shown in the table shows that generating vector representations for spike proteins using ProtVec takes more time compared to the time taken for generating vector representations using 3-mers. The time difference in seconds is minimal in terms of pre-processing aligned and unaligned sequences where the vector representation method remains the same. It is to be noted at this point that we considered the metadata in chronological order based on the 'Collection Data' column. The sequences for the spike proteins have been retrieved from GISAID based on the 'gisaid_epi_isl' column. The variant information is extracted from the 'PANGO Lineage' column and serves as labels. In order to present our results, we consider the records from the oldest record till the first occurrence of the variant to be the training data and the records from the first occurrence of a variant till the most recent record to be the test data.

Table 1– Time taken by various pre-processing methods

Pre-processing method	Time (seconds)
ProtVec unaligned	7318.8668
3-mer unaligned	88.4325
ProtVec aligned	7347.6524
3-mer aligned	89.2525

As previously discussed in the Methods section, we consider four novelty detection algorithms as part of this study: (i) Elliptic Envelope, (ii) Isolation Forest, (iii) Local Outlier Factor, and (iv) One Class SVM. In order to determine the performance of the novelty detection algorithms, we use an accuracy metric formulated as the fraction of the total number of records that the novelty method correctly classifies as a 'novelty' out of the total number of records containing the variant labels in the test data.

Table 2– Accuracies for identification of variant B.1.1.7 using various novelty detection methods

B.1.351						
Method	**ProtVec unaligned**	**ProtVec aligned**	**ProtVec unaligned + PCA**	**ProtVec aligned + PCA**	**3-mer unaligned + PCA**	**3-mer aligned + PCA**
EllipticEnvelope	0.1622	0.1622	0.1622	0.1622	0	0
IsolationForest	0.3108	0.3108	0.3108	0.2635	0	0
LocalOutlierFactor	**0.7297**	0.6824	0.3784	0.2973	0.0068	0.0068
OneClassSVM	**0.7568**	0.5203	0.6892	0.5405	0.6419	0.6554
Time (seconds)	3.4829	3.2946	2.3084	2.2796	65.4902	62.3319

This gives a value between 0 and 1 where values closer to 1 indicate better performance of the respective novelty detection algorithm. We first consider the default parameters for all of the above algorithms in sci-kit learn and apply it to the COVID-19 training data for the B.1.1.7 and B.1.351 variants respectively. Table 1 and Table 2 show our results on the B.1.1.7 and B.1.351 variants respectively. The cells highlighted in yellow indicate accuracies of the respective novelty detection algorithm above 70% that show relatively better performance depending on the pre-processing method. The results of both B.1.1.7 and B.1.351 indicate that One Class SVM gives the best performance. However, Table 2 and Table 3 only provide an overview of the results using default parameters of all novelty detection algorithms and at this point, it is important to also consider tuning parameters. Table 4 shows the results obtained after fine tuning parameters on OneClassSVM on ProtVec unaligned. We show results on varying the type of the kernel used for SVM as well as different values of the nu parameter that provides upper and lower bounds on the errors during training and support vectors respectively.

The default kernel type is 'rbf' and the default nu value is 0.5 in sklearn OneClassSVM (accuracies on the default values have been indicated in Table 2 and Table 3).

We test out all combinations for kernel = {'linear', 'poly', 'rbf', 'sigmoid'} and nu = {0.2, 0.4, 0.6, 0.8}. Higher accuracies above 70% are highlighted in yellow.

Table 3– Accuracies for identification of variant B.1.351 using various novelty detection methods

B.1.1.7				
Parameter	nu=0.2	nu=0.4	nu=0.6	nu=0.8
kernel='linear'	0.5155	0.5567	0.6005	**0.7526**
kernel='poly'	0.5206	0.5232	0.6082	**0.7526**
kernel='rbf'	0.2474	0.6881	**0.7552**	**0.7964**
kernel='sigmoid'	0.5541	0.518	0.6005	**0.7526**
B.1.351				
Parameter	nu=0.2	nu=0.4	nu=0.6	nu=0.8
kernel='linear'	0.4527	0.4662	0.4932	0.6081
kernel='poly'	0.4662	0.4662	0.5	0.6149
kernel='rbf'	0.3378	0.6689	**0.7703**	**0.8243**
kernel='sigmoid'	0.4527	0.4662	0.4932	0.6081

Discussion

The main aim of our study was applying a set of novelty detection algorithms on COVID-19 spike proteins in order to detect new variants.

It is challenging to apply novelty detection algorithms to such a scenario because: (i) there is a notion of time with 'new' variants occurring with the passage of time and (ii) there is no scope of validating the dataset since we consider the testing dataset records begin with the first occurrence of the novel variant.

We applied a set of novelty detection algorithms to the problem of detecting new COVID-19 variants. As discussed in the Results section, we show the accuracies on default parameters of these algorithms in Table 2 and Table 3 for the B.1.1.7 and B.1.351 variants respectively. Our results show that LocalOutlierFactor and OneClassSVM both perform reasonably well on ProtVec unaligned, with OneClassSVM performing slightly better. It is interesting to note that aligning sequences does not provide a significant edge in terms of performance in the context of this problem. This can be explained in terms of adding extra 'noise' since we are adding the 'unknown' vector everytime '-' is encountered. However, this is not a problem with the 3-mer approach since the presence of '-' is discounted while using CountVectorizer from sklearn. Overall for both variants, ProtVec gives reasonable performance compared to the 3-mer approach. This can be explained based on the fact that ProtVec uses vectors for each 'word' to convey meaning while constructing the vector for the entire sequence. In essence, we formulated a computer science approach using ProtVec, but there is contextual meaning behind each of the protein 3-grams from a biological perspective while using ProtVec. In addition, while using ProtVec, we add vector representations of 3-grams based on their occurrence, so we are also taking frequency into account. The final vector for the sequence thus contains the contextual as well as frequency information. However, the 3-mer vector is similar to a bag-of-words approach where we simply group together counts of 3-mer proteins based on frequency of occurrence. Using ProtVec gave us 100-dimensional vector representations of sequences while using the 3-mer approach, we obtained 3110-dimensional vector representations, which were difficult to apply to the novelty detection algorithms. We have used PCA as a means of reducing dimensions, the results of which are also indicated in Table 2 and Table 3. However, applying PCA to ProtVec did not yield any beneficial results in terms of accuracy. For the 3-mer approach, PCA was the only way to reduce the number of features and apply the novelty detection algorithms as there was significant computational overhead otherwise.

Table 4– Accuracies for parameters of OneClassSVM on ProtVec unaligned

B.1.1.7						
Method	**ProtVec unaligned**	**ProtVec aligned**	**ProtVec unaligned + PCA**	**ProtVec aligned + PCA**	**3-mer unaligned + PCA**	**3-mer aligned + PCA**
EllipticEnvelope	0.0876	0.0902	0.0902	0.0902	0.0129	0
IsolationForest	0.1959	0.1959	0.2191	0.2397	0	0
LocalOutlierFactor	**0.7242**	**0.7345**	0.2861	0.2113	0.0052	0
OneClassSVM	**0.7448**	0.482	**0.7139**	0.4639	0.5799	0.5799
Time (seconds)	3.7431	3.7604	2.3089	2.3318	63.3319	64.1851

As shown in Table 1, even though the 3-mer approach is faster in terms of pre-processing compared to ProtVec, it is beneficial to use ProtVec in order to get better results in terms of accuracies in the context of novelty detection in COVID-19 spike protein sequences. It is important to note at this point that the process of reducing features using PCA in the 3-mer method takes significant time compared to the ProtVec method while applying the novelty detection algorithms, and hence, the time taken by ProtVec is faster during the novelty detection phase itself. We then show results of fine-tuning parameters on OneClassSVM for ProtVec unaligned in Table 4. Our results indicate that the 'rbf' kernel provides the best accuracies at higher values of the 'nu' parameter for both variants. We obtained the highest accuracies of 79.64% and 82.43% for the B.1.1.7 and B.1.351 variants respectively.

We believe that our study will provide useful insights to aid researchers in various countries to formulate mitigation strategies or study vaccines and medicines based on the new variants. This method is suitable for finding variants in real-time and can indicate the presence of new variants on a specific day. However, our work comes with certain limitations. We only focus on detecting novel variants as they appear, not create labels for their clinical significance. Future studies and extensions of this work could also consider integrating explanations on a biological basis and explaining our results from a biological and medical perspective.

Conclusions

In this paper, we study the performance of a set of novelty detection algorithms and apply them to the detection of new COVID-19 variants. We show results on the B.1.1.7 and B.1.351 variants and obtain accuracies of 79.64% and 82.43% on B.1.17 and B.1.351 using OneClassSVM on ProtVec unaligned COVID-19 spike protein sequences with fine-tuned parameters. We believe that our study can help understand the virus further during the ongoing pandemic.

Code

The code for this paper is available at this link: https://github.com/sayantanibasu/covid19-variants .

Acknowledgements

This project has been funded by the Jump ARCHES endowment through the Health Care Engineering Systems Center. This work uses data from GISAID (https://www.gisaid.org). We would like to acknowledge all contributing laboratories that have contributed their COVID-19 sequence data to GISAID.

References

[1] COVID-19 Map - Johns Hopkins Coronavirus Resource Center, *Johns Hopkins Coronavirus Resource Center*. (2020). https://coronavirus.jhu.edu/map.html (accessed April 22, 2021).

[2] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, and U.R. Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, *Computers in Biology and Medicine*. **121** (2020) 103792.

[3] E.E.-D. Hemdan, M.A. Shouman, and M.E. Karar, Covidx-net: A framework of deep learning classifiers to diagnose COVID-19 in x-ray images, *ArXiv Preprint ArXiv:2003.11055*. (2020).

[4] H. Jelodar, Y. Wang, R. Orji, and S. Huang, Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach, *IEEE Journal of Biomedical and Health Informatics*. **24** (2020) 2733–2742.

[5] Y. Li, T. Grandison, P. Silveyra, A. Douraghy, X. Guan, T. Kieselbach, C. Li, and H. Zhang, Jennifer for COVID-19: An NLP-powered chatbot built for the people and by the people to combat misinformation, (2020).

[6] S. Basu, and R.H. Campbell, Going by the numbers: Learning and modeling COVID-19 disease dynamics, *Chaos, Solitons & Fractals*. **138** (2020) 110140.

[7] S. Basu, A study of the dynamics and genetics of COVID-19 through machine learning, MS Thesis, University of Illinois at Urbana-Champaign, 2020.

[8] S.-C. Wu, Progress and concept for COVID-19 vaccine development, *Biotechnology Journal*. (2020).

[9] G.S. Randhawa, M.P. Soltysiak, H. El Roz, C.P. de Souza, K.A. Hill, and L. Kari, Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study, *Plos One*. **15** (2020) e0232391.

[10] G. Wang, W. Wei, J. Jiang, C. Ning, H. Chen, J. Huang, B. Liang, N. Zang, Y. Liao, R. Chen, and others, Application of a long short-term memory neural network: a burgeoning method of deep learning in forecasting HIV incidence in Guangxi, China, *Epidemiology & Infection*. **147** (2019).

[11] R. Yin, E. Luusua, J. Dabrowski, Y. Zhang, and C.K. Kwoh, Tempel: time-series mutation prediction of influenza A viruses via attention-based recurrent neural networks, *Bioinformatics*. **36** (2020) 2697–2704.

[12] E. Asgari, and M.R.K. Mofrad, Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics, PLOS ONE. 10 (2015) e0141287. doi:10.1371/journal.pone.0141287.

[13] P.J. Rousseeuw, and K.V. Driessen, A fast algorithm for the minimum covariance determinant estimator, *Technometrics*. **41** (1999) 212–223.

[14] F.T. Liu, K.M. Ting, and Z.-H. Zhou, Isolation forest, in: 2008 Eighth IEEE International Conference on Data Mining, 2008: pp. 413–422.

[15] F.T. Liu, K.M. Ting, and Z.-H. Zhou, Isolation-based anomaly detection, *ACM Transactions on Knowledge Discovery from Data (TKDD)*. **6** (2012) 1–39.

[16] M.M. Breunig, H.-P. Kriegel, R.T. Ng, and J. Sander, LOF: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000: pp. 93–104.

[17] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson, Estimating the support of a high-dimensional distribution, *Neural Computation*. **13** (2001) 1443–1471.

[18] M.E. Tipping, and C.M. Bishop, Probabilistic principal component analysis, Journal of the Royal Statistical Society: Series B (Statistical Methodology). 61 (1999) 611–622.

[19] K. Katoh, K. Misawa, K. Kuma, and T. Miyata, MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform, *Nucleic Acids Research*. **30** (2002) 3059–3066.

[20] E. Asgari, protVec_100d_3grams.csv, in: Replication Data for: Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics, Harvard Dataverse, 2015. doi:10.7910/DVN/JMFHTN/CVPAUK.

[21] Genomic epidemiology of novel coronavirus - Global subsampling, (2021). https://nextstrain.org/ncov/global (accessed April 16, 2021).

[22] J. Hadfield, C. Megill, S.M. Bell, J. Huddleston, B. Potter, C. Callender, P. Sagulenko, T. Bedford, and R.A. Neher, Nextstrain: real-time tracking of pathogen evolution, Bioinformatics. 34 (2018) 4121–4123.

[23] W. McKinney, pandas: a foundational Python library for data analysis and statistics, *Python for High Performance and Scientific Computing*. **14** (2011) 1–9.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and others, Scikit-learn: Machine learning in Python, *The Journal of Machine Learning Research*. **12** (2011) 2825–2830.

Address for correspondence

Name: Roy H. Campbell
Email: rhc@illinois.edu