PCD2Vec: A Poisson Correction Distance Based Approach for Viral Host Classification

Sarwan Ali, Taslim Murad, Murray Patterson

IJCNN 2023

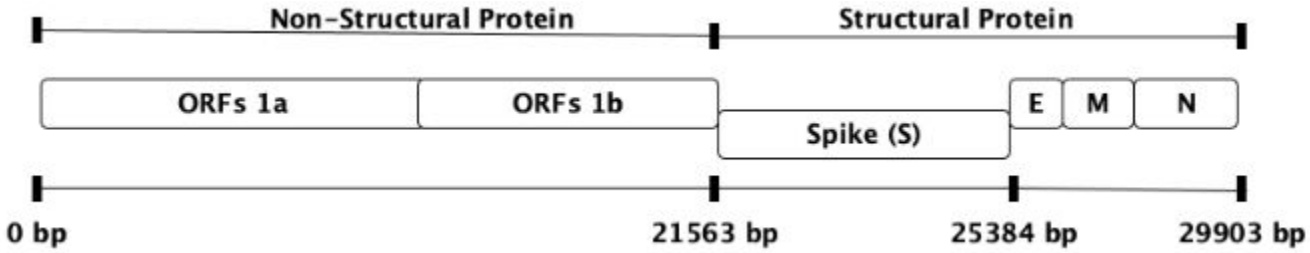
Georgia State University

Table of Content

Motivation

- Coronaviruses are membrane-enveloped, non-segmented positive-strand RNA viruses belonging to the Coronaviridae family.
- They are well-known for causing pandemic,
  - SARS-CoV (severe acute respiratory syndrome coronavirus) in 2003.
  - MERS-CoV (Middle East respiratory syndrome coronavirus) in 2012.
  - SARS-CoV-2 (severe acute respiratory syndrome coronavirus 2) in 2019.
- They have infected various organisms like animals, humans, birds etc.
- Their genomic sequence analysis can provide information about the genetic diversity and dynamic of the virus which is helpful in designing the prevention mechanisms e.g vaccines, drugs etc.
  - Analysis like viral infected host classification.
- Machine learning (ML) models are good option for doing sequence host classification,
  - However they requires the inputs to be in numerical form.
  - Therefore, efficient and effective techniques are needed to convert bio-sequences into numerical form.

Motivation



- Spike protein region gives sufficient information for viral host classification,
  - ➢ It is used to attach to the host cell membrane.
- Therefore only use spike sequence (rather than full genome) to perform host classification.

Introduction

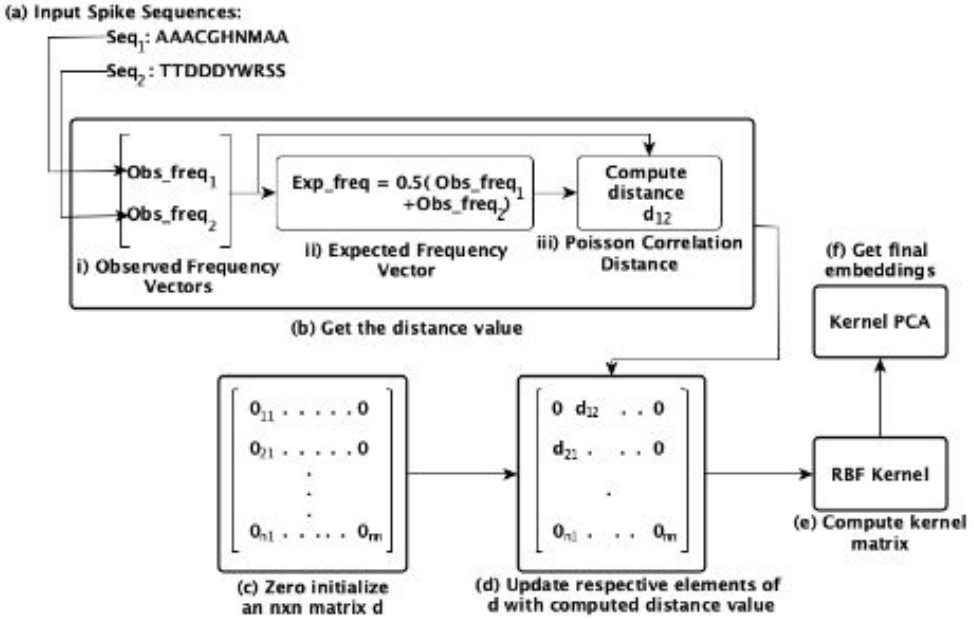
- We formulate a method to convert spike protein sequences into numerical form by using the Poisson Correction Distance (PCD) concept to enable ML model based host classification.
- PCD is a measure of the difference in amino acid composition between two protein sequences.
  - The theoretical basis for this distance measure is the Poisson distribution, which models the number of events occurring in a fixed interval of time.
  - The PCD formula uses the observed and expected frequencies of each amino acid in two sequences and the Poisson distribution to calculate the distance between the sequences.
  - This distance is a good measure because it takes into account both the observed and expected frequencies of each amino acid in the sequences, and it also considers the variability in the frequencies of the amino acids by using the Poisson distribution.

Existing Works

- Many works exits to perform bio-sequence analysis and some of them are summarized as follow,

Methods	**Drawbacks**
One-hot encoding	Sparsity and curse of dimensionality
Phylogenetic approaches	Not scalable (computationally expensive)
K-mer based methods	Sparsity and computationally expensive

Proposed System – Workflow



$$d = 2 \times exp\_freq \times (\ln(\frac{obs\_freq1}{exp\_freq}) + \ln(\frac{obs\_freq2}{exp\_freq}))$$

Proposed System – Algorithm

Algorithm 1 The algorithm for PCD-based embedding generation for spike sequences.

Input: Set of Spike Sequences ($seqs$)
Output: Embeddings

$distances \leftarrow zeros(len(seqs), len(seqs))$
for i in $|seqs| - 1$ **do**
  for j in $(i + 1, |seqs|)$ **do** ▷ Upper Triangle Only
    /* Compute the observed frequencies of each amino acid */
    $obs\_freq1 \leftarrow$ AMINOACIDFREQ($seqs[i]$)
    $obs\_freq2 \leftarrow$ AMINOACIDFREQ($seqs[j]$)
    /* Compute the expected frequencies */
    $exp\_freq \leftarrow 0.5 \times (obs\_freq1 + obs\_freq2)$
    $d \leftarrow 0$
    for k in $|20|$ **do** ▷ 20 Amino Acids
      if $exp\_freq[k] > 0$ **then**
        $\epsilon \leftarrow 0.0001$ ▷ to avoid divided by 0 error
        $obs\_freq1[k] \leftarrow obs\_freq1[k] + \epsilon$
        $obs\_freq2[k] \leftarrow obs\_freq2[k] + \epsilon$
        Freq_1 $\leftarrow ln(\frac{obs\_freq1[k]}{exp\_freq[k]})$
        Freq_2 $\leftarrow ln(\frac{obs\_freq2[k]}{exp\_freq[k]})$
        Freq $\leftarrow$ Freq_1 + Freq_2
        $d \leftarrow$ d + 2 $\times$ exp_freq[k] $\times$ Freq
      end if
    end for
    $distances[i, j] \leftarrow d$
    $distances[j, i] \leftarrow d$
  end for
end for
$kernelMatrix \leftarrow$ RBFKERNEL(distances)
$Embedding \leftarrow$ KERNELPCA($kernelMatrix$)

Proposed System – Properties

- We also proof that the our distance matrix holds 3 properties, which are,
  - **Triangle inequality:** The triangle inequality property ensures that the distance between two points via a third point is always equal to or greater than the direct distance between the two points. This property ensures that the distance metric is consistent and well-defined
  - **Symmetry:** The symmetry property ensures that the distance between two points is the same regardless of the order in which the points are considered. This property ensures that the distance metric is consistent and unbiased.
  - **Non-negativity:** The non-negativity property ensures that the distance between any two points is always non-negative, i.e., it is either zero or a positive number. This property ensures that the distance metric is well-defined and that it has a clear meaning.
- These properties ensure that the distance matrix generated is a valid distance metric that can be used for various machine-learning tasks.

Proposed System – Contributions

- Our contributions to this paper are as follows:
  - **Efficient Prediction:** We show that coronavirus hosts can be efficiently predicted using spike sequences only.
  - **Incorporation of biological knowledge:** Our method to generate a low-dimensional embedding, based on the Poisson correction distance (PCD), better captures the biological relationships between the spike protein sequences in the classification task, which general distance measures / representation learning methods may not consider.
  - **Use of RBF kernel:** We used the RBF kernel to project the data into high dimensional space, which has been proven to perform well in non-linear classification tasks and is often used in the analysis of biological sequences.
  - **Use of kernel PCA:** We used Kernel PCA, which allows us to perform dimensionality reduction while preserving the non-linear structure of the data. This can lead to better separation between the different classes and improved classification performance.
  - **Theoretical proofs for three properties:** We provide theoretical proofs for the triangle inequality, symmetry, and non-negativity properties to ensure the validity of the distance metric used in our method, which can add confidence to the results obtained from our method.

Experimental Setup – Dataset Statistics

- The dataset used for host classification is summarized in the table below,

Host	Count	Host	Count
human	957	pangolin	5
swine	785	duck	3
chicken	309	chimpanzee	3
camel	265	goose	2
bat	181	beluga Whale	2
cat	57	falcon	1
civet	5	-	-
Total	2575		

TABLE I: Host (class label) distribution in data.

Experimental Setup – ML Models

- For classification in the experiments we used the following ML models:
  - Support Vector Machine (SVM)
  - Naive Bayes (NB)
  - Multilayer Perceptron (MLP)
  - k-Nearest Neighbor (k-NN) (where $k = 3$)
  - Random Forest (RF)
  - Logistic Regression (LR).

Experimental Setup – Baselines

- One-Hot Encoding (OHE) [1]
- One-Hot Encoding + PCA
- Ridge Regression [2]
- Lasso Regression [3]
- Autoencoder [4]
- Poincaré Embeddings [5]
- String Kernel [6]
- Protein Bert [7]

Results & Discussion

- PCD2Vec is outperforming,
  - All feature engineering-based baselines (OHE, Lasso Regression, PCA, Ridge Regression).
  - The NN-based Autoencoder method.
  - String kernel.
  - Huge improvement over Poincaré Embeddings.
  - Improvement over pre-trained model-based method Protein bert.

Method	Classifier	Accuracy		Precision		Recall		F1 Weigh.		F1 Macro		ROC AUC	
		Avg.	Var.	Avg.	Var.	Avg.	Var.	Avg.	Var.	Avg.	Var.	Avg.	Var.
PCA	SVM	0.90	0.0001	0.92	0.0002	0.90	0.0001	0.88	0.0003	0.82	0.0002	0.90	0.0125
	NB	0.86	0.0002	0.93	0.0001	0.86	0.0004	0.87	0.0002	0.77	0.0005	0.93	0.0114
	MLP	0.91	0.0001	0.91	0.0001	0.91	0.0002	0.90	0.0001	0.84	0.0004	0.90	0.0212
	KNN	0.94	0.0003	0.94	0.0001	0.94	0.0003	0.93	0.0002	0.86	0.0001	0.92	0.0152
	RF	0.95	0.0002	0.96	0.0001	0.95	0.0003	0.95	0.0002	0.95	0.0002	0.97	0.0099
	LR	0.91	0.0003	0.91	0.0001	0.91	0.0002	0.90	0.0001	0.84	0.0002	0.90	0.0104
AutoEncoder	SVM	0.92	0.0003	0.92	0.0003	0.92	0.0002	0.90	0.0001	0.82	0.0002	0.89	0.0099
	NB	0.78	0.0001	0.87	0.0002	0.78	0.0001	0.80	0.0001	0.69	0.0003	0.89	0.0107
	MLP	0.93	0.0001	0.94	0.0003	0.93	0.0001	0.93	0.0002	0.89	0.0002	0.94	0.0097
	KNN	0.93	0.0004	0.94	0.0002	0.93	0.0004	0.93	0.0001	0.86	0.0002	0.93	0.0092
	RF	0.95	0.0002	0.96	0.0001	0.95	0.0002	0.95	0.0003	0.94	0.0001	0.97	0.0114
	LR	0.91	0.0002	0.91	0.0004	0.91	0.0002	0.90	0.0002	0.80	0.0004	0.88	0.0012
Lasso Regression	SVM	0.95	0.0003	0.96	0.0004	0.95	0.0003	0.95	0.0001	0.94	0.0002	0.97	0.0094
	NB	0.92	0.0001	0.95	0.0005	0.92	0.0003	0.92	0.0004	0.91	0.0005	0.96	0.0059
	MLP	0.94	0.0003	0.95	0.0005	0.94	0.0002	0.94	0.0004	0.90	0.0005	0.94	0.0098
	KNN	0.93	0.0001	0.92	0.0003	0.93	0.0002	0.92	0.0003	0.88	0.0002	0.92	0.0047
	RF	0.95	0.0001	0.96	0.0001	0.95	0.0002	0.95	0.0001	0.95	0.0003	0.97	0.0098
	LR	0.94	0.0003	0.94	0.0004	0.94	0.0003	0.93	0.0001	0.91	0.0002	0.94	0.0025
Ridge Regression	SVM	0.95	0.0001	0.96	0.0002	0.95	0.0001	0.95	0.0005	0.94	0.0004	0.97	0.0075
	NB	0.94	0.0004	0.96	0.0005	0.94	0.0001	0.94	0.0002	0.92	0.0005	0.96	0.0071
	MLP	0.93	0.0002	0.94	0.0003	0.93	0.0002	0.93	0.0005	0.88	0.0006	0.93	0.0064
	KNN	0.92	0.0001	0.92	0.0002	0.92	0.0001	0.92	0.0005	0.86	0.0001	0.91	0.0027
	RF	0.95	0.0002	0.96	0.0004	0.95	0.0002	0.95	0.0003	0.94	0.0005	0.97	0.0028
	LR	0.94	0.0003	0.94	0.0002	0.94	0.0003	0.94	0.0001	0.91	0.0002	0.95	0.0046
OHE	SVM	0.95	0.0001	0.96	0.0003	0.95	0.0002	0.95	0.0005	0.94	0.0004	0.97	0.0078
	NB	0.94	0.0002	0.96	0.0001	0.94	0.0002	0.94	0.0005	0.93	0.0001	0.97	0.0051
	MLP	0.94	0.0003	0.94	0.0002	0.94	0.0003	0.93	0.0004	0.89	0.0001	0.94	0.0069
	KNN	0.93	0.0001	0.95	0.0003	0.93	0.0002	0.93	0.0001	0.90	0.0004	0.95	0.0074
	RF	0.95	0.0004	0.96	0.0002	0.95	0.0004	0.95	0.0003	0.94	0.0001	0.97	0.0028
	LR	0.94	0.0002	0.95	0.0004	0.94	0.0002	0.94	0.0001	0.93	0.0003	0.96	0.0088
String Kernel	SVM	0.94	0.0007	0.95	0.0002	0.94	0.0007	0.94	0.0014	0.90	0.0006	0.95	0.0019
	NB	0.69	0.0019	0.86	0.0017	0.69	0.0019	0.72	0.0011	0.70	0.0019	0.86	0.0004
	MLP	0.82	0.0011	0.81	0.0030	0.82	0.0031	0.81	0.0025	0.44	0.0040	0.71	0.0023
	KNN	0.93	0.0007	0.93	0.0055	0.93	0.0097	0.92	0.0092	0.61	0.0023	0.82	0.0030
	RF	0.95	0.0010	0.96	0.0025	0.95	0.0010	0.95	0.0063	0.91	0.0059	0.95	0.0083
	LR	0.94	0.0008	0.95	0.0017	0.94	0.0071	0.94	0.0018	0.90	0.0067	0.95	0.0015
Poincaré Embedding	SVM	0.39	0.0001	0.34	0.0001	0.39	0.0001	0.33	0.0001	0.10	0.0008	0.52	0.0004
	NB	0.74	0.0001	0.71	0.0008	0.74	0.0001	0.72	0.0004	0.30	0.0022	0.64	0.0005
	MLP	0.64	0.0001	0.56	0.0002	0.64	0.0001	0.59	0.0002	0.21	0.0008	0.58	0.0001
	KNN	0.60	0.0002	0.57	0.0002	0.60	0.0002	0.57	0.0003	0.21	0.0005	0.58	0.0001
	RF	0.78	0.0005	0.74	0.0009	0.78	0.0005	0.73	0.0008	0.31	0.0019	0.64	0.0004
	LR	0.34	0.0003	0.30	0.0013	0.34	0.0003	0.27	0.0003	0.08	0.0001	0.50	0.0000
Protein Bert	-	0.92	0.0004	0.93	0.0002	0.92	0.0003	0.91	0.0001	0.86	0.0002	0.92	0.0003
PCD2Vec (Ours)	SVM	0.87	0.0098	0.90	0.0508	0.87	0.0098	0.86	0.0193	0.74	0.1030	0.87	0.0505
	NB	0.68	0.0470	0.87	0.0227	0.68	0.0470	0.71	0.0450	0.75	0.0724	0.90	0.0304
	MLP	0.84	0.0209	0.85	0.0219	0.84	0.0209	0.84	0.0236	0.64	0.0839	0.79	0.0392
	KNN	0.93	0.0107	0.94	0.0153	0.93	0.0107	0.93	0.0136	0.70	0.0663	0.90	0.0430
	RF	**0.97**	0.0085	**0.97**	0.0099	**0.96**	0.0085	**0.96**	0.0090	**0.98**	0.0842	**0.99**	0.0454
	LR	0.86	0.0116	0.87	0.0580	0.86	0.0116	0.84	0.0248	0.62	0.0802	0.80	0.0395

TABLE II: Average and variance results (of 5 runs) for different methods. The best average values are shown in bold.

Conclusion

- In this paper, we presented a novel method for predicting the host specificity of coronaviruses by analyzing spike protein sequences.
- Our method involves the use of Poisson correction distance, radial basis function kernel, and kernel PCA to generate low-dimensional embeddings of the spike protein sequences.
- Future work will focus on refining and improving our method and testing it on larger and more diverse datasets.

References

[1] Kiril Kuzmin et al. Machine learning methods accurately predict host specificity of coronaviruses based on spike sequences alone. Biochemical and Biophysical Research Communications, 533(3):553–558, 2020.

[2] Gary C McDonald. Ridge regression. Wiley Interdisciplinary Reviews: Computational Statistics, 1(1):93–100, 2009

[3] J Ranstam and JA Cook. Lasso regression. Journal of British Surgery,105(10):1348–1348, 2018

[4] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In International conference on machine learning, pages 478–487, 2016

[5] Maximillian Nickel and Douwe Kiela. Poincar´e embeddings for learning hierarchical representations. Advances in neural information processing systems, 30, 2017

[6] Sarwan Ali, Bikram Sahoo, Muhammad Asad Khan, Alexander Zelikovsky, Imdad Ullah Khan, and Murray Patterson.Efficient approximate kernel based spike sequence classification.IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2022

[7]Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. ProteinBERT: a universal deep-learning Model of protein sequence and function. Bioinformatics , 38(8):2102–2110, 2022.