-ORIGINAL ARTICLE-

An analysis of k-mer frequency features with SVM and CNN for viral subtyping classification

Un análisis de atributos de frecuencia de k-mer con SVM y CNN para la clasificación de subtipos de virus

Vicente Enrique Machaca Arceda[1]

[1]*Departamento de Ingeniería y Matemáticas, Universidad La Salle, Peru*
vmachacaa@ulasalle.edu.pe

Abstract

Viral subtyping classification is very relevant for the appropriate diagnosis and treatment of illnesses. The most used tools are based on alignment-based methods, nevertheless, they are becoming too slow due to the increase of genomic data; for that reason, alignment-free methods have emerged as an alternative. In this work, we analyzed four alignment-free algorithms: two methods use k-mer frequencies (Kameris and Castor-KRFE); the third method used a frequency chaos game representation of a DNA with CNNs; and the last one processes DNA sequences as a digital signal (ML-DSP). From the comparison, Kameris and Castor-KRFE outperformed the rest, followed by the method based on CNNs.

Keywords: CNN, genome, viral subtyping, k-mer, Kameris, Castor, ML-DSP.

Resumen

La clasificación de subtipos de virus es muy importante para el diagnóstico y tratamiento de enfermedades. Las herramientas más utilizadas dependen de algoritmos basados en alineamiento, sin embargo, estos métodos, se estan volviendo muy lentos con el crecimiento de información. Por esta razón, estan emergiendo nuevos métodos no basados en alineamiento. En este trabajo, se han analizado cuatro algoritmos no basados en alineamiento: dos de ellos, se basan en las frecuencias de k-mer (Kameris y Castor-KRFE); el tercer método utiliza a *frequency chaos game representation* del ADN junto con CNNs; el ultimo método, procesa el ADN como si fuera una señal digital (ML-DSP). Kameris y Castor-KRFE obtuvieron los mejores resultados seguidos por el método basado en CNNs.

Palabras claves: CNN, genoma, subtipos de virus, k-mer, Kameris, Castor, ML-DSP.

1 Introduction

In Virology, viral subtyping refers to a fundamental unit of virus nomenclature within a defined specie [1]. It is very important for taxonomic studies and disease treatment.

Each virus, could be divided into different subtypes, for instance, Human Immunodeficiency Viruses (HIV) are classified into two types: HIV-1 and HIV-2. HIV-1 is related to viruses in chimpanzees and gorillas, while HIV-2 is related to viruses of the sooty mangabey primate. Furthermore, HIV-1 is divided into groups: M, N, O, and P. Inside M group, there are subtypes A, B, C, D, E, F, G, H, I, K, K, and L that are related to the location where they emerged. Also, there are Circulating Recombinant Forms (CRFs) that refers to the different combination between HIV-1 subtypes [2].

The appropriate virus subtype classification is relevant for disease treatment, for example, one of the obstacles to the HIV treatment is its high genetic variability [2]. The HIV-1's genome is diverged by about 15% [3]. Furthermore, some authors claim that a correct classification of HIV-1 subtypes is required for clinical management [4].

There are several methods for viral classification, they could be grouped into alignment-based, organism-specifics, and alignment-free methods. [5]. One of the most popular method used for genome similarity is the alignment-based method BLAST [6]. Furthermore, alignment-based is not suitable for viral genome classification because of their genetic variation (recombination, shuffling, and horizontal gene transfer events) [7]. Also, we need prior knowledge to adjust some parameters in order to get good results [8]. In addition, the majority of the alignment-based and organism-specifics are proprietary and they require laboratories and servers to transmit the sequence, this expose sensitive information [1].

In this study, we analyzed the performance of k-mer frequency like feature vectors with Support Vector Machines (SVM) and Convolutional Neural Networks (CNN) for viral subtyping classification. We compared Kameris [1], Castor-KRFE [5], and an own implementation, using Frequency Chaos Game Representation (FCGR) with CNN. Additionally, we include a method based on digital signal processing [9].

The work is structured as follows: in Section 2, we present the most relevant related works, Section 3 describes the methods, parameters and data used in order to replicate the experiment, Section 4 describes the results, Section 5 comments the results, Section 6 shows the conclusions and finally in Section 7, we show the future work.

2 Viral subtyping classification

The most popular tools for viral subtyping classification are REGA [10], SCUEAL [11] and USEARCH [12], all of them are based on alignment-based methods and phylogenetics analysis. Furthermore, there are pair-wise alignment-based methods like BLAST [6], TurboBLAST [13], ScalaBLAST [14] and multiple sequence alignment (CLUSTAL [15]).

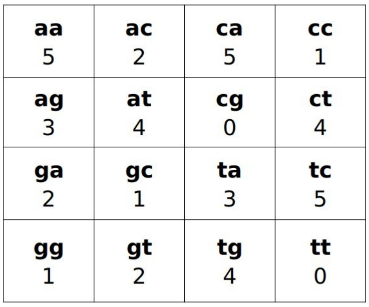
On the other hand, there are alignment-free methods. They compute a feature vector and then, they use a distance algorithm or a machine learning model in order to classify a genome [16, 17]. The viral (subtype) classification problem in alignment-free methods takes as input, a set of sequences $s = \{s_1, s_2, ...s_{n-1}, s_n\}$, where s_i is a string of characters $\omega = \{A, C, G, T\}$ and y representing the class of each sequence. Some alignment-free methods are Comet[18], Castor [19], VirFinder [20], a data warehouse with machine learning algorithms [21], an open source framework Kameris [1] and Castor-KRFE [5].

Blaisdell [22] proposed one of the first methods using k-mer frequencies. These k-mer frequencies have been used for phylogenetic studies and machine learning models during the past decades [23, 24, 25, 26]. Other researches used k-mer for sequence prediction [27] or sequence assembly [28]. The k-mers can be viewed as sub-strings of a sequence, for example, for the sequence $s = \{A, C, T, G, A, C\}$, the 3-mers set is: $\{ACT, CTG, TGA, GAC\}$, and the 2-mers set: $\{AC, CT, TG, GA, AC\}$. The alignment-free methods base on k-mer frequencies compute the frequency of each element in a k-mer set, they usually experiment with different values of k in order to get the best k for a given dataset.

Moreover, some researches have used Chaos Game Representations (CGR) as a genomics signature along with machine learning models; For instance, CGR genomic signatures for HIV-1 sub-typing [29], CGR for the analysis of Circulating Recombinant Forms (CRFs) of HIV-1 [30], classification of the genotypes of Human Papilloma Virus (HPV) [31], CGR for classification of viral pathogens [32], and more recently, CGR for rapid classification of COVID-19 [33].

2.1 Kameris

Proposed by Solis-Reyes [1] in 2018, they computed the k-mer frequencies using a Frequency Chaos Game Representation (FCGR), this FCGR is derived from the Chaos Game Representation (CGR) [34]. A FCGR could be represented in Fig. 1, each cell in the matrix represents a specific k-mer, the first cell, for example, represents the 2-mer *AA*, and it appears 5 times in a sequence. Besides, in Fig. 2, we present how each cell of a FCGR matrix represents a specific k-mer, the *C* quadrant sub-divided into *CA, CC, CG, CT* and the *CT* quadrant sub-divided into *ACT, CCT, GCT, TCT*, the division depends on the number of levels or the *k* value in k-mer, in this way each k-mer has a unique position in a FCGR matrix. Another example is shown in Fig. 3, here we present the FCGR matrix for one (1-mer), two (2-mer) and three (3-mer) levels, in the case of $L = 2$ we could see all the possible 2-mers of a sequence. Also, in Fig. 1, we represent the number of times each k-mer appears in a sequence. In this way, it is easy to store the k-mer frequencies in the FCGR.



aa 5	**ac** 2	**ca** 5	**cc** 1
ag 3	**at** 4	**cg** 0	**ct** 4
ga 2	**gc** 1	**ta** 3	**tc** 5
gg 1	**gt** 2	**tg** 4	**tt** 0

Figure 1: A FCGR k-mer example, each k-mer is represented as a cell in the matrix, and the frequency of each k-mer is represented as the pixel value.

After processing the FCGR as a feature vector, it is applied a Single Value Decomposition (SVD) for dimensionality reduction. Then, they used a Support Vector Machine (SVM). The authors evaluated their work for the classification of HIV-1 virus subtypes.

2.2 Castor-KRFE

Proposed by Lebatteux et al. [5] in 2019, they computed the k-mer frequencies in a different form than Kameris. For example, from a set of sequences $S = \{s_1, s_2, ..., s_n\}$, the authors obtain the k-mers frequencies presented only in the dataset. For example

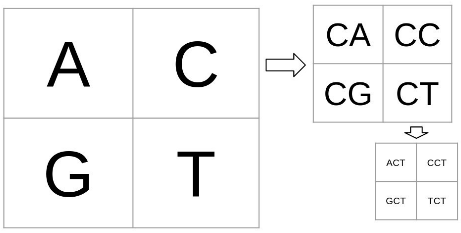
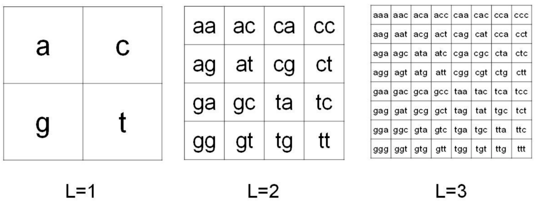


Figure 2: A FCGR matrix. The *G* quadrant sub-divided into the corresponding G-endings and the *TG* quadrant sub-divided into the corresponding *TG-endings*.



a	c
g	t

L=1

aa	ac	ca	cc
ag	at	cg	ct
ga	gc	ta	tc
gg	gt	tg	tt

L=2

aaa	aac	aca	acc	caa	cac	cca	ccc
aag	aat	acg	act	cag	cat	cca	cct
aga	agc	ata	atc	cga	cgc	cta	ctc
agg	agt	atg	att	cgg	cgt	ctg	ctt
gaa	gac	gca	gcc	taa	tac	tca	tcc
gag	gat	gcg	gct	tag	tat	tgc	tct
gga	ggc	gta	gtc	tga	tgc	tta	ttc
ggg	ggt	gtg	gtt	tgg	tgt	ttg	ttt

L=3

Figure 3: A FCGR example for one, two and three levels. Each cell represents a specific k-mer.

if *S*={*ACTGC, TTGCG* }, the *3-mers* will be: { *ACT, CTG, TGC, TTG, GCG* } avoiding repeatition, then the feature vector will be: $\{1,1,2,1,1\}$, we noticed that the 3-mer *TGC* appears twice in the dataset. In contrast to Kameris, Castor-KRFE obtains a feature vector smaller than Kameris. Moreover, Castor-KRFE uses Recursive Feature Elimination (RFE) based on SVM to eliminate features. After that, they used SVM as a classifier. Also, they evaluated their proposal in different viral datasets obtaining the best k and the number of features for each dataset.

2.3 FCGR and CNN

In this method, we used Frequency Chaos Game Representation (FCGR) of a DNA sequence along with Convolutional Neural Network (CNN) to classify virus subtyping.

The problem of represent DNA sequences as images, was first proposed by H.J. Jeffrey et al. [34]. A formal definition is denote by Eq. (1), for a DNA sequence $s_1, s_2, ..., s_n$, the corresponding CGR sequence $(X_n) = (x_n, y_n)$ is given by:

$$X_0 = (\frac{1}{2}, \frac{1}{2}), X_n = \frac{X_{n-1} + W}{2} \tag{1}$$

where W represents the coordinates of the corners of the unit square $A = (0,0)$; $C = (0,1)$; $G = (1,1)$; $T = (1,0)$ if s_n is a, c, g, t respectively. In this form, we plotted a point for each value of X_n. For example, in Fig. 4 (left), we plotted the points according to the sequence $s = T, G, C, A$. Moreover, in Fig. 4 (right), we plotted the complete CGR of a HIV-1 whole genome. Jeffrey et al. [34] demonstrated that, each specimen have different fractal patterns in its CGR.

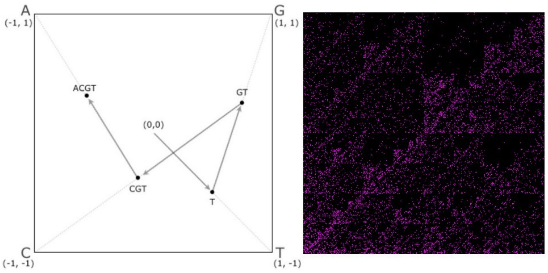


Figure 4: Chaos game representation of a DNA sequence. Left: CGR point of sequence: $s = T, G, C, A$. Right: CGR of a HIV-1 whole genome

Then, Wang et al.[35] proved that CGR's patterns are completely determined by frequencies of oligonucleotides of all lengths (a deep study of CGR is presented by Joseph et al. [36]). These frequencies of oligonucleotides are commonly known as k-mer frequencies and they could be represented as a FCGR (Fig. 1). Moreover, these frequencies could be interpreted as pixel intensities to represent an image (Fig. 5).

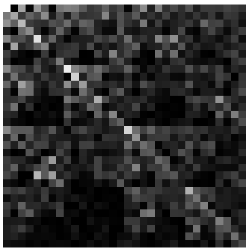


Figure 5: A FCGR (k=5) of a HIV-1 genome.

Finally, we used a Convolutional Neural Network (CNN) to classify a sequence. We used the CNN's architecture proposed by Fabijańska et al. [37] (Fig. 7) and other two small models detailed in Fig. 6. We did not use Max Pooling, because the images from a FCGR with $k = 5$ were small ($32x32$).

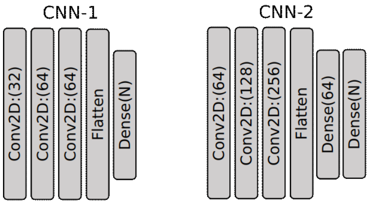


Figure 6: CNN's architecture proposed.

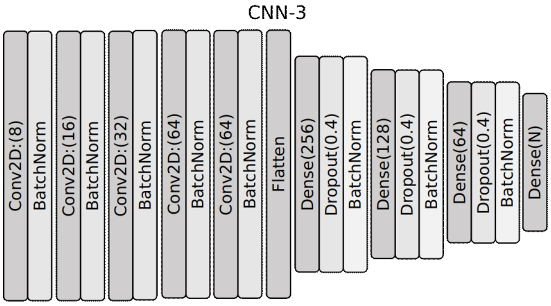


Figure 7: CNN's architecture proposed.

2.4 ML-DSP

Machine Learning with Digital Signal Processing (ML-DSP), is proposed by Randhawa et al. [9]. In this case, a DNA sequence is mapped to numbers, where each base $T/C = 1$ and $A/G = -1$. Then, the authors applied a Discrete Fourier Transform (DFT) and its magnitude spectrum. Finally, a Pearson Correlation Coefficient is computed. Then, they used a Support Vector Machine (SVM) to train the model.

2.5 Summary

We compared the performance of Kameris [1], Castor-KRFE [5], CNNs and ML-DSP. The first two methods, use k-mer frequencies, but they differ in the way they computed it. Kameris computed all the 4^k possible k-mers using a FCGR matrix, it has an efficient way of getting the frequencies they but got a big feature vector. In contrast, Castor-KRFE computed just the k-mers frequencies presented in the training set, they computed it inefficiently but, they got a feature vector smaller than Kameris. Moreover, Kameris used SVD for dimensionality reduction meanwhile Castor-KRFE use RFE for feature elimination. The third method, uses a FCGR images and CNNs. Finally, the last method (ML-DSP), consider a DNA sequence as a digital signal.

In Table 1, we presented the methods described before, from now on we are going to use the nomenclature of this table for each method.

3 Materials and Method

We used the datasets from Castor[1], they are the first 10 rows of Table 2. HBVGENCG is a group of sequences of Hepatits-B virus; HIVGRPCG, HIVSUBCG and HIVSUBPOL are sequences of HIV-1; INFSUBHA, INFSUBMP and INSUBFNA are sequences from Influenza virus; EBOSPECG, RHISPECG and HPVGENCG are sequences of Ebola, Rhinovirus and Human papillomavirus respectively.

Table 1: Methods used in this research.

Method name	Description
Kameris-SVD	Kameris with dimensionality reduction SVD. The feature vector was reduced to 10%.
Kameris	Kameris without dimensionality reduction.
Castor-KRFE	Castor with feature elimination RFE.
Castor	Castor without feature elimination.
CNN-1, CNN-2 and CNN-3	Convolutional Neural Networks (CNN) with FCGR as images. We used three architectures: CNN-1, CNN-2 and CNN-3 (Fig. 6 and 7).
ML-DSP	Machine Learning Digital Signal Processing (ML-DSP) uses Discrete Fourier transform and Pearson Correlation Coefficient.

Also, we used a group of dataset proposed by Randhawa et al. [9], these datasets are the last 9 rows of Table 2.

For CNN models, we used Adam optimizer, mini-batch size of 128 and 100 epochs. Moreover, We used seed of 1 for random numbers in weights initialization.

For processing time analysis (Table 6), we used an Intel i5 processor of 1.9 GHz, and 8 GBytes of memory RAM.

4 Results

In this section, we show a comparison between the methods proposed (Kameris, Castor, ML-DSP, CNN-1, CNN-2 and CNN-3). Additionally, we compared the processing time of these methods against BLAST. In the experiments, *k* refers to k-mer.

4.1 F-score and feature vector size

Kameris-SVD/Kameris and Castor-KRFE/Castor are very similar so, we compared them exhaustively. We evaluated the f-score metric with and without dimensionality reduction and feature elimination. A comparison of f-score is presented in Fig. 8, we used k-mers, ranging from k=1 to k=9. We concluded that Kameris, and Castor without applying dimensionality reduction and feature elimination have the same results. In contrast, when we apply SVD to Kameris, the f-score decreases slightly, but when we apply RFE

[1] http://castor.bioinfo.uqam.ca/

Table 2: The datasets used in the experiments.

Data sets	Average seq. length	No. of classes	No. of instances
HIVGRPCG	9164	4	76
HIVSUBCG	8992	18	597
HIVSUBPOL	1211	28	1352
INFSUBHA	1719	2	10825
INFSUBMP	759	2	21421
INSUBFNA	1416	2	10715
EBOSPECG	18917	5	751
HBVGENCG	3189	8	230
RHISPECG	369	3	1316
HPVGENCG	7610	3	125
Primates	16626	2	148
Dengue	10595	4	4721
Protists	31712	3	159
Fungi	49178	3	224
Plants	277931	2	174
Amphibians	17530	3	290
Insects	15689	7	898
3classes	16292	3	2170
Vertebrates	16806	5	4322

to Castor, the f-score decreases more than Kameris.

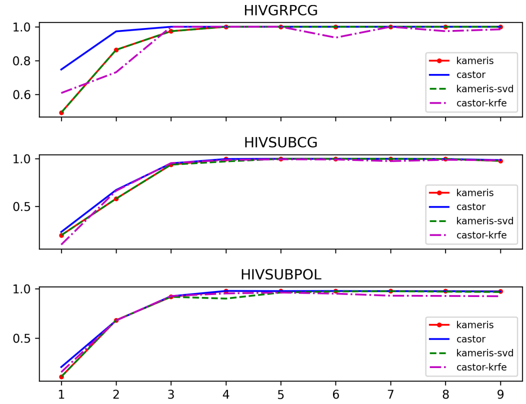


Figure 8: F-score (y-axis) comparison between Kameris-SVD/Kameris and Castor-KRFE/Castor for k-mers ranging from k=1 to k=9 (x-axis).

Moreover, we evaluated how the feature vector size of Kameris and Castor increases according to k (Fig. 9). As we can see, since $k = 6$ Kameris's feature vector is too big, it is bigger than Castor's feature vector, and that is because Kameris's feature vector size is 4^k (it considers all possible k-mers) meanwhile Castor just considers the k-mers presented in the dataset. Also, we evaluated how the feature vector size of Kameris-SVD and Castor-KRFE increases according to k (Fig. 10), Kameris-SVD's feature vector size is bigger than Castor-KRFE's feature vector since 5-mers.

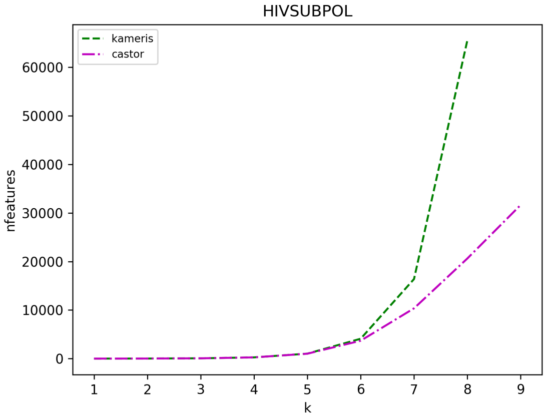


Figure 9: Feature vector size of Kameris and Castor according to k . We used HIVSUBPOL dataset for this experiment

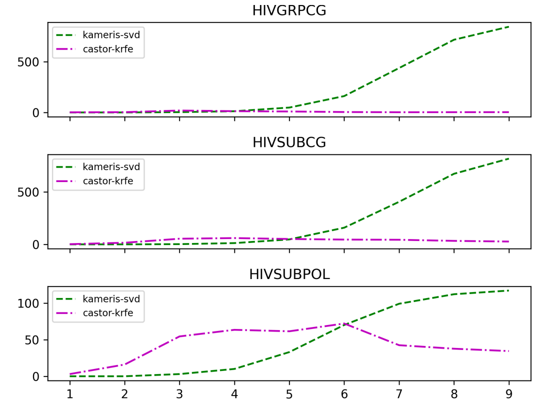


Figure 10: Feature vector size of Kameris-SVD and Castor-KRFE according to k . We used HIVSUBPOL dataset for this experiment.

In Table 3, we show the best f-score that we got with the smaller value of k, for Kameris-SVD and Castor-KRFE. Kameris-SVD slightly outperform Castor-KRFE, meanwhile, Castor-KRFE's feature vector is smaller than Kameris' feature vector. Furthermore, in Table 4, we show the f-score of Kameris-SVD and Castor-KRFE with 5-mers and 10-fold validation.

In Table 5, we show the f-score of the methods proposed. In this case, we used Kameris and Castor without dimensionality reduction and feature elimination because, they got better results in this way. We noticed that ML-DSP got poor results meanwhile, Kameris and Castor got the best results.

4.2 Processing time

Also, we compared the processing time of each algorithm against BLAST. We choose BLAST because

Table 3: The best f-score that we got with the smaller value of k. Also, the number of features is presented.

		Kameris-SVD	
Dataset	(k-mer)	f-score	nfeatures
HIVGRPCG	4	1.0000	12
HIVSUBCG	5	**0.9983**	47
HIVSUBPOL	7	**0.9761**	99
		Castor-KRFE	
Dataset	(k-mer)	f-score	nfeatures
HIVGRPCG	3	1.0000	19
HIVSUBCG	5	0.9937	51
HIVSUBPOL	5	0.9629	65

Table 4: F-score and standard deviation of Kameris-SVD and Castor-KRFE (5-mer) with 10-fold validation.

	Kameris-SVD		Castor-KRFE	
Dataset	f-score	sd	f-score	sd
HIVGRPCG	1.000	0.000	1.000	0.000
HIVSUBCG	**0.999**	0.003	0.992	0.008
HIVSUBPOL	**0.965**	0.022	0.961	0.023

this is the most used method for viral subtyping classification and similarity analysis. For example, REGA compares a DNA sequence (query) against 87 other sequences (HIV-1 samples), then according to the phylogenetic analysis, the method classifies a sequence. A simple query on these tools could take some minutes but, we can not measure exactly this time because this is web service.

For Kameris, Castor, ML-DSP, and CNN (CNN-1, CNN-2 and CNN-3), we measured the processing time to get the feature vector; for BLAST, we measured the processing time to align the query sequence against 76 other sequences (the smallest dataset have 76 instances); we used the *bioblast* library. For all the experiments, we run each method 10 times and then, we took the average time.

In Table 6, we show a comparison of the processing time. Castor, ML-DSP and CNN (CNN-1, CNN-2 and CNN-3) got the best results, meanwhile BLAST got poor results against the others.

4.3 Limitations

In this study, we noticed that the processing time to compute k-mer frequency could take too much time since 7-mer. Additionally, bigger genomes need bigger k-mers [5]. In this context, we need efficient algorithms in order to compute k-mer frequency. Moreover, there are datasets with a small number of samples that are not appropriated to use in machine learning.

5 Discussion

Kameris and Castor without their dimensional reduction and feature elimination are almost the same, despite Castor got a small feature vector (the extra attributes of Kameris are zero values). Meanwhile, if we applied SVD and RFE to Kameris and Castor respectively, the f-score slightly is reduced. Moreover, according to the results in Table 3 and 4, kameris-SVD outperformed Castor-KRFE.

In Table 5, ML-DSP got poor results, meanwhile the authors presented good results in their paper [9]. We noticed that, they compute Pearson Correlation Coefficients (PCC) over the whole dataset, then they split it in train and test. Nevertheless, it is not correct, they should have split the dataset into train and test, and then they should have computed the PCC only on training set.

In Table 6, we compared the processing time against BLAST. When we use BLAST for viral classification, we need to align a query sequence against others to find the most similar, and then we could infers the class of a sequence. Nevertheless, this method depends strongly on the amount of instances in a dataset. Currently, the datasets in Bioinformatics are increasing every day, so BLAST will be slower.

6 Conclusions

In this work, we evaluated four methods for viral subtyping classification, based on alignment-free algorithms. Two methods use k-mer frequencies (Kameris and Castor), the third method uses CNNs, and finally the last method treats a DNA sequence as a signal.

Kameris-SVD and Castor-KRFE, compute k-mer frequencies, then they used SVD and RFE for dimensionality reduction and feature elimination. Finally, they used a SVM as a classifier. We noticed that Kameris-SVD outperformed slightly Castor-KRFE. Moreover, if we did not use SVD and RFE, they got the same f-score. Additionally, Castor-KRFE got a smaller feature vector than Kameris-SVD, it is very important considering the huge amount of data in Bioinformatic.

ML-DSP processed a DNA sequence as a digital signal, but this method got poor results against CNNs, Kameris, and Castor. Kameris and Castor without SVD and RFE got the best accuracy, but they are

Table 5: F-score of the methods proposed (5-mer). We put a '-' for the methods that couldn't converge.

Dataset	CNN-1	CNN-2	CNN-3	ML-DSP	Kameris	Castor
Primates	1	1	0.964	1	1	1
Dengue	1	1	1	0.998	1	1
Protists	1	1	0.882	0.805	1	1
Fingi	0.978	1	0.923	0.859	1	1
Plants	0.882	0.85	0.917	0.78	0.882	0.972
Amphibians	0.983	1	0.982	0.967	1	1
Insects	0.961	0.949	0.966	0.893	0.994	0.994
3classes	1	1	1	1	1	1
Vertebrates	0.995	0.998	0.995	0.985	0.998	0.998
HIVGRPCG	0.909	0.913	-	0.838	1	1
HIVSUBCG	0.962	0.975	0.978	0.686	1	1
HIVSUBPOL	0.981	0.981	0.981	-	0.993	0.993
INFSUBHA	1	1	1	1	1	1
INFSUBMP	0.986	0.989	0.989	0.935	0.989	0.988
EBOSPECG	1	1	1	0.982	1	1
HBVGENCG	1	1	1	0.773	1	1
RHISPECG	1	1	1	1	1	1
HPVGENCG	1	1	1	1	1	1

Table 6: Time processing in milliseconds, for each method to process a DNA sequence, we used 5-mer for each method. CNN stands for the time processing to compute FCGR images in CNN-1, CNN2 and CNN-3.

Dataset	BLAST	Kameris	Castor	ML-DSP	CNN	Sequence length
Primates	0.609	0.052	0.006	0.005	0.005	16499
Dengue	0.175	0.033	0.003	0.004	0.004	10313
Protists	0.227	0.066	0.007	0.006	0.006	24932
Fungi	0.402	0.240	0.026	0.026	0.017	190834
Plants	2.160	0.299	0.034	0.043	0.029	103830
Amphibians	0.397	0.057	0.006	0.006	0.005	16101
Insects	0.135	0.045	0.005	0.005	0.004	14711
3classes	0.110	0.044	0.005	0.004	0.004	14496
Vertebrates	0.417	0.049	0.006	0.005	0.005	16442
HIVGRPCG	0.229	0.027	0.003	0.003	0.003	8654
HIVSUBCG	0.351	0.026	0.003	0.003	0.003	8589
HIVSUBPOL	0.130	0.004	0.001	0.001	0.001	1017
INFSUBHA	0.095	0.006	0.001	0.001	0.002	1704
INFSUBMP	0.086	0.003	0.000	0.001	0.001	759
INSUBFNA	0.081	0.005	0.001	0.001	0.001	1413
EBOSPECG	0.209	0.058	0.006	0.006	0.005	18828
HBVGENCG	0.146	0.011	0.001	0.001	0.002	3182
RHISPECG	0.073	0.001	0.000	0.000	0.001	369
HPVGENCG	0.115	0.022	0.002	0.002	0.003	7100

followed by CNNs.

Also, we compared the processing time against BLAST. In this case, Castor, ML-DSP and CNNs got the best results. Moreover, there is a big difference against BLAST.

7 Future work

K-mer frequencies are good descriptors for viral subtyping classification. Nevertheless, despite the good performance, they have a high cost for $k > 5$; large genomes require higher values of k. For that reason, we are going to evaluate another algorithms like the inverted-index technique to compute k-mer frequencies. Moreover, we are going to evaluate the use of another deep learning technique for viral classification.

Competing interests

The authors have declared that no competing interests exist.

Authors' contribution

V.E. Machaca conceived the idea, wrote the program, developed the experiments, analyzed the results, wrote and revised the manuscript.

References

[1] S. Solis-Reyes, M. Avino, A. Poon, and L. Kari, "An open-source k-mer based machine learning tool for fast and accurate subtyping of hiv-1 genomes," *PloS one*, vol. 13, no. 11, 2018.

[2] P. M. Sharp and B. H. Hahn, "Origins of hiv and the aids pandemic," *Cold Spring Harbor perspectives in medicine*, vol. 1, no. 1, p. a006841, 2011.

[3] J. B. Joy, R. H. Liang, T. Nguyen, R. M. McCloskey, and A. F. Poon, "Origin and evolution of human immunodeficiency viruses," in *Global Virology I-Identifying and Investigating Viral Diseases*, pp. 587–611, Springer, 2015.

[4] N. Clumeck, A. Pozniak, F. Raffi, and E. E. Committee, "European aids clinical society (eacs) guidelines for the clinical management and treatment of hiv-infected adults," *HIV medicine*, vol. 9, no. 2, pp. 65–71, 2008.

[5] D. Lebatteux, A. M. Remita, and A. B. Diallo, "Toward an alignment-free method for feature extraction and accurate classification of viral sequences," *Journal of Computational Biology*, vol. 26, no. 6, pp. 519–535, 2019.

[6] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped blast and psi-blast: a new generation of protein database search programs," *Nucleic acids research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[7] S. Duffy, L. A. Shackelton, and E. C. Holmes, "Rates of evolutionary change in viruses: patterns and determinants," *Nature Reviews Genetics*, vol. 9, no. 4, pp. 267–276, 2008.

[8] A. Zielezinski, S. Vinga, J. Almeida, and W. M. Karlowski, "Alignment-free sequence comparison: benefits, applications, and tools," *Genome biology*, vol. 18, no. 1, p. 186, 2017.

[9] G. S. Randhawa, K. A. Hill, and L. Kari, "Ml-dsp: Machine learning with digital signal processing for ultrafast, accurate, and scalable genome classification at all taxonomic levels," *BMC genomics*, vol. 20, no. 1, p. 267, 2019.

[10] T. De Oliveira, K. Deforche, S. Cassol, M. Salminen, D. Paraskevis, C. Seebregts, J. Snoeck, E. J. Van Rensburg, A. M. Wensing, D. A. Van De Vijver, *et al.*, "An automated genotyping system for analysis of hiv-1 and other microbial sequences," *Bioinformatics*, vol. 21, no. 19, pp. 3797–3800, 2005.

[11] S. L. K. Pond, D. Posada, E. Stawiski, C. Chappey, A. F. Poon, G. Hughes, E. Fearnhill, M. B. Gravenor, A. J. L. Brown, and S. D. Frost, "An evolutionary model-based algorithm for accurate phylogenetic breakpoint mapping and subtype prediction in hiv-1," *PLoS computational biology*, vol. 5, no. 11, 2009.

[12] R. C. Edgar, "Search and clustering orders of magnitude faster than blast," *Bioinformatics*, vol. 26, no. 19, pp. 2460–2461, 2010.

[13] R. D. Bjornson, A. Sherman, S. B. Weston, N. Willard, and J. Wing, "Turboblast (r): A parallel implementation of blast built on the turbohub," in *ipdps*, p. 0183, IEEE, 2002.

[14] C. Oehmen and J. Nieplocha, "Scalablast: a scalable implementation of blast for high-performance data-intensive bioinformatics analysis," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, pp. 740–749, 2006.

[15] D. G. Higgins and P. M. Sharp, "Clustal: a package for performing multiple sequence alignment on a micro-computer," *Gene*, vol. 73, no. 1, pp. 237–244, 1988.

[16] S. Vinga, "Alignment-free methods in computational biology," 2014.

[17] Z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification," *ACM Sigkdd Explorations Newsletter*, vol. 12, no. 1, pp. 40–48, 2010.

[18] D. Struck, G. Lawyer, A.-M. Ternes, J.-C. Schmit, and D. P. Bercoff, "Comet: adaptive context-based modeling for ultrafast hiv-1 subtype identification," *Nucleic acids research*, vol. 42, no. 18, pp. e144–e144, 2014.

[19] M. A. Remita, A. Halioui, A. A. M. Diouara, B. Daigle, G. Kiani, and A. B. Diallo, "A machine learning approach for viral genome classification," *BMC bioinformatics*, vol. 18, no. 1, p. 208, 2017.

[20] J. Ren, N. A. Ahlgren, Y. Y. Lu, J. A. Fuhrman, and F. Sun, "Virfinder: a novel k-mer based tool for identifying viral sequences from assembled metagenomic data," *Microbiome*, vol. 5, no. 1, p. 69, 2017.

[21] J. C. F. Silva, T. F. Carvalho, M. F. Basso, M. Deguchi, W. A. Pereira, R. R. Sobrinho, P. M. Vidigal, O. J. Brustolini, F. F. Silva, M. Dal-Bianco, *et al.*, "Geminivirus data warehouse: a database enriched with machine learning approaches," *BMC bioinformatics*, vol. 18, no. 1, p. 240, 2017.

[22] B. E. Blaisdell, "A measure of the similarity of sets of sequences not requiring sequence alignment," *Proceedings of the National Academy of Sciences*, vol. 83, no. 14, pp. 5155–5159, 1986.

[23] X. Liu, L. Wan, J. Li, G. Reinert, M. S. Waterman, and F. Sun, "New powerful statistics for alignment-free sequence comparison under a pattern transfer model," *Journal of theoretical biology*, vol. 284, no. 1, pp. 106–116, 2011.

[24] R. H. Chan, T. H. Chan, H. M. Yeung, and R. W. Wang, "Composition vector method based on maximum entropy principle for sequence comparison," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 9, no. 1, pp. 79–87, 2011.

[25] G. E. Sims, S.-R. Jun, G. A. Wu, and S.-H. Kim, "Alignment-free genome comparison with feature frequency profiles (ffp) and optimal resolutions," *Proceedings of the National Academy of Sciences*, vol. 106, no. 8, pp. 2677–2682, 2009.

[26] I. Ulitsky, D. Burstein, T. Tuller, and B. Chor, "The average common substring approach to phylogenomic

reconstruction," *Journal of Computational Biology*, vol. 13, no. 2, pp. 336–350, 2006.

[27] M. Ghandi, D. Lee, M. Mohammad-Noori, and M. A. Beer, "Enhanced regulatory sequence prediction using gapped k-mer features," *PLoS computational biology*, vol. 10, no. 7, 2014.

[28] R. Chikhi and P. Medvedev, "Informed and automated k-mer size selection for genome assembly," *Bioinformatics*, vol. 30, no. 1, pp. 31–37, 2014.

[29] A. Pandit and S. Sinha, "Using genomic signatures for hiv-1 sub-typing," *BMC bioinformatics*, vol. 11, no. S1, p. S26, 2010.

[30] A. Bansiwal, *Analysis of Circulating Recombinant Forms (CRFs) of HIV-1 using Chaos Game Representation (CGR).* PhD thesis, IISER M, 2014.

[31] W. Tanchotsrinon, C. Lursinsap, and Y. Poovorawan, "A high performance prediction of hpv genotypes by chaos game representation and singular value decomposition," *BMC bioinformatics*, vol. 16, no. 1, p. 71, 2015.

[32] E. Adetiba, J. A. Badejo, S. Thakur, V. O. Matthews, M. O. Adebiyi, and E. F. Adebiyi, "Experimental investigation of frequency chaos game representation for in silico and accurate classification of viral pathogens from genomic sequences," in *International Conference on Bioinformatics and Biomedical Engineering*, pp. 155–164, Springer, 2017.

[33] 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," *bioRxiv*, 2020.

[34] H. J. Jeffrey, "Chaos game representation of gene structure," *Nucleic acids research*, vol. 18, no. 8, pp. 2163–2170, 1990.

[35] Y. Wang, K. Hill, S. Singh, and L. Kari, "The spectrum of genomic signatures: from dinucleotides to chaos game representation," *Gene*, vol. 346, pp. 173–185, 2005.

[36] J. Joseph and R. Sasikumar, "Chaos game representation for comparison of whole genomes," *BMC bioinformatics*, vol. 7, no. 1, p. 243, 2006.

[37] A. Fabijańska and S. Grabowski, "Viral genome deep classifier," *IEEE Access*, vol. 7, pp. 81297–81307, 2019.

Citation: V.E. Machaca. *An analysis of k-mer frequency features with SVM and CNN for viral subtyping classification*. Journal of Computer Science & Technology, vol. 20, no. 2, pp. 99–107, 2020.

DOI: 10.24215/16666038.20.e11.

Received: March 21, 2020 **Accepted:** October 5, 2020.