Preprint of:
Nawaz, M. S., Fournier-Viger, P., Nawaz, S., Zhu, H., Yun U. (2024). SPM4GAC: SPM based approach for genome analysis and classification of macromolecules. International Journal of Biological Macromolecules (BIOMAC). Elsevier, Volume 266, Part 2, May 2024, article 130984
DOI: 10.1016/j.ijbiomac.2024.130984

SPM4GAC: SPM based approach for genome analysis and classification

M. Saqib Nawaz[1] · **Philippe Fournier-Viger**[1,*] · **Shoaib Nawaz**[2] · **Haowei Zhu**[1] · **Unil Yun**[3]

the date of receipt and acceptance should be inserted later

[1]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
[2]Department of Pharmacy, The University of Lahore, Sargodha Campus, Pakistan
[3]Sejong University, Seoul, Korea
E-mail: msaqibnawaz@szu.edu.cn, philfv@szu.edu.cn, shoaib.nawaz@pharm.uol.edu.pk, zhuhaowei2020@email.szu.edu.cn, yunei@sejong.ac.kr
* Corresponding author

Abstract Genome sequence analysis and classification play critical roles in properly understanding an organism's main characteristics, functionalities, and changing (evolving) nature. However, the rapid expansion of genomic data makes genome sequence analysis and classification a challenging task due to the high computational requirements, proper management, and understanding of genomic data. In this paper, we present SPM4GAC, a sequential pattern mining (SPM)-based framework to analyze and classify the genome sequences of viruses. First, a large dataset containing the genome sequences of various RNA viruses is developed and transformed into a suitable format. On the transformed dataset, algorithms for SPM are used to identify frequent sequential patterns of nucleotide bases. The obtained frequent sequential patterns of bases are then used as features to classify different viruses. Ten classifiers are employed, and their performance is assessed by using several evaluation measures. Finally, a performance comparison of SPM4GAC with state-of-the-art methods for genome sequence classification/detection reveals that SPM4GAC performs better than those methods.

Keywords Genomes · RNA Virus · Classification · Sequential pattern Mining · Nucleotides.

1 Introduction

A genome in molecular biology is an encoded sequence containing nucleotide bases (Poor & Yaghoobi, 2019) and represents the complete set of an entire organism's genetic material. Genomes can now be sequenced at a rapid pace, thanks to advanced sequence technology techniques, and can be shared on public repositories such as GenBank (Sayers et al., 2020), NGDC (Members & Partners, 2023) and GISAID (Kalia, Saberwal, & Sharma, 2021). However, the rapid growth in size and complexity of genomics data has created new challenges for analyzing and interpreting large biological datasets. The vast and intricate biological data are beyond the capacity of traditional approaches. Most taxonomical genome classification tools use alignment-based approaches. For example, BLAST (Altschul, Gish, Miller, Myers, & Lipman, 1990) and FASTA (Pearson, 1994) are widely used and regarded as the references for sequence analysis and classification. However, these methods have certain limitations, such as requiring more time and memory when very large genome sequences are aligned, not operating well in low sequence identity scenarios, and the alignment results are dependent on various presumptions and criteria. (Zielezinski, Vinga, Almeida, & Karlowski, 2017; Vinga, 2013).

Similarly, genome classification approaches that are based on *k-mers* or minimizers (Zielezinski et al., 2017; Ye, Siddle, Park, & Sabeti, 2019; Roberts, Hunt, Yorke, Bolanos, & Delcher, 2004) require wider regions with high similarity, which can yield lower recall and precision. It is now imperative to overcome the problems of storage, management, and processing of massive genomic data and be able to extract effective information while ensuring that this information carries a true biological meaning. Moreover, the emergence of the COVID-19 (F. Wu et al., 2020; Ahamad et al., 2020) pandemic in late 2019 and its impact showed the world that efficient

computational approaches for genome analysis and classification, as well as well-organized databases and search engines are required to face pandemics. In addition, a wide range of viruses, particularly harmful ones, are emerging frequently around the world. We believe that the huge amount of genomic data and their easy availability offer a rich resource for frequent pattern mining and machine learning (ML) to be used as alternatives to extract important information from biological sequences and in the development of generic classification systems.

The main aims of this study are to: (1) develop a large corpus for genome sequences of RNA viruses; and (2) investigate how sequential pattern mining (SPM) (Fournier-Viger, Lin, Kiran, Koh, & Thomas, 2017) is useful for the reliable classification and detection of genomes. SPM in genomics can offer new and meaningful insights for organism behavior and can identify hidden information that can be of importance to the biological community and thus aid in accelerating biological research. SPM has been used in the past to find interesting hidden nucleotide bases and amino acid patterns in genome sequences and their relationships with each other (Nawaz, Fournier-Viger, Shojaee, & Fujita, 2021; Nawaz, Fournier-Viger, Aslam, et al., 2023), analyze gene expression (Zihayat, Davoudi, & An, 2017), mine DNA datasets for maximal contiguous frequent patterns (Karim, Rashid, Jeong, & Choi, 2012), mine rare cardiovascular disease symptom rules (Iqbal et al., 2022), discover motifs in DNA sequences (Hsu, Chen, Hsu, & Liu, 2006), predict protein function (Wang, Shang, & Li, 2008), discover patterns for gene interactions and their characterizations (Cellier et al., 2013), interpret patterns extracted from DNA microarrays (Sallaberry, Pecheur, Bringay, Roche, & Teisseire, 2011) and to recognize protein folds (Exarchos, Papaloukas, Lampros, & Fotiadis, 2008). However, no studies have yet been done that show the usefulness of frequent sequential patterns for the efficient classification/detection of genome sequences without providing the whole sequences.

Some recently published ML and deep learning (DL)-based studies (Ali et al., 2021; Alshayeji, Sindhu, & Abed, 2023; Arslan & Arslan, 2021; Arslan, 2021a, 2021b; Lopez-Rincon et al., 2021; Naeem, Mabrouk, Marzouk, & Eldosoky, 2021; Randhawa et al., 2020; Ahmed & Jeon, 2022; Singh et al., 2021; El-Dosuky, Soliman, & Hassanien, 2021; Jing et al., 2020; Gunasekaran et al., 2021; Mateos, Balboa, Easteal, Eyras, & Patel, 2021; Dlamini et al., 2021) classify and detect genome sequences that belong to various viruses and species. Most of these studies focused on finding important features (such as CpG-based features, *k-mers*-based features, representative genomic sequences, intrinsic genomic signatures, intrinsic dinucleotide genomic signatures, and biomarkers) and used different kinds of encoding and embedding techniques (e.g. one-hot and dictionary encoding, *k-mers*, label encoding, discrete Fourier and discrete Cosine transform, and moment invariant). As far as we are aware, there is currently no published study or method on the use of pattern mining for whole genome sequence classification, especially for harmful viruses. The following are the main contributions made in this paper:

- A dataset is developed that contains genome sequences belonging to various RNA virus types. The genome sequences were taken from NCBI's GenBank, and they were preprocessed to make them suitable for applying SPM approaches. The developed corpus can serve as an experimental testbed and benchmark for

ML and DL tasks in genomics. The dataset is provided at: github.com/saqibdola/SPM4GAC.

– Based on the analysis of nucleotide bases in genome sequences, we design a framework called SPM4GAC, (SPM for Genome Analysis and Classification), which offers a genome sequence analysis and classification/detection approach. SPM4GAC classifies genome sequences based on frequent sequential patterns of nucleotide bases that are discovered by using SPM algorithms. For classification, ten classifiers are used, and comprehensive experiments are carried out by using various metrics to investigate the efficacy of the developed classification system.

The proposed SPM4GAC approach was applied to a developed corpus that contains genomes sequences (taken from GenBank) of 15 RNA viruses. Obtained results by using the proposed SPM4GAC framework on the developed corpus indicate that using frequent sequential patterns of nucleotide bases as features provides superior classification performance than using all nucleotide bases in the whole genome sequences. It was found that overall, two text-based classifiers and one integer-based classifier performed well. Compared to integer-based, the text-based classifiers were slow during the training and testing phases. Obtained results by comparing the performance of SPM4GAC with state-of-the-art (SOTA) genome sequence classification/detection methods demonstrated that SPMF4GAC performs better than SOTA methods.

The remaining four sections of the paper are: Section 2 provides the related work on the use of ML and DL for genome analysis and classification/detection. Section 3 presents the details of the created dataset and the proposed SPM4GAC framework. The results are presented and discussed in Section 4. Lastly, the paper is concluded in Section 5.

2 Related Work

ML and DL approaches have been used for the classification and prediction of genome sequences. For example, the studies (Arslan, 2021b; Arslan & Arslan, 2021; Arslan, 2021a) used CpG as features to classify SARS-CoV-2 genomes. The genome classification system of (Naeem et al., 2021) used various discrete transforms and moment invariants-based features. (Lopez-Rincon et al., 2021) used a convolutional neural network (CNN) combined with explainable AI techniques to discover representative genomic sequences. (Randhawa et al., 2020) first identified intrinsic genomic signature and then used them with a ML-based alignment-free (AF) classification approach. (Ahmed & Jeon, 2022) used various standard ML algorithms to classify four viruses (Ebola, MERS, SARS-CoV-1 and SARS-CoV-2) genomes. Biomarkers, based on three-base periodicity, were used in (Singh et al., 2021) for the classification of genomes. (El-Dosuky et al., 2021) used a CNN, with a cockroach optimization algorithm, to classify viruses genomes. (Alshayeji et al., 2023) implemented one method based on *k-mers* and their frequencies for the identification of viral genomes in human DNA sequences.

Some studies have considered a gene in place of a whole genome sequence for the classification of viruses. For example, the classification approach in (Ali et al., 2021)

for Spike (S) protein sequences of SARS-CoV-2 variants was based on *k-mers* (for the generation of feature vector) and kernel approximation (for computing pairwise similarity among sequences). Similarly, one-hot encoding was used on S sequences to classify various coronaviruses (Kuzmin et al., 2020). S-PDB (Nawaz, Fournier-Viger, & He, 2022) used the amino acid sequences of S protein structures, obtained from Protein Data Bank (PDB), and their aligned amino acids and aligned secondary structure elements for classification. Another study (Nawaz, Fournier-Viger, He, & Zhang, 2023) presented the PSAC-PDB framework to analyze and classify protein structures in the PDB. (Qiang, Xu, Fang, Liu, & Kou, 2020) collected the S protein sequences of 2,666 coronaviruses. Three feature encoding algorithms were used to obtain important features from S sequences that were used to train various random forest classifiers.

(Gunasekaran et al., 2021) employed various DL methods (CNN, CNN-LSTM, and CNN-Bidirectional LSTM) to classify DNA sequences that were encoded using label and *k-mer* encoding. Another classification approach (Dlamini et al., 2021) is based on analyzing the intrinsic dinucleotide genomic signatures. The genome sequences of eight pathogenic species were first transformed into dinucleotide relative frequencies, that were then classified using the XGBoost model. (Mateos et al., 2021) developed PACIFIC, a DL-based classifier, to detect various RNA viruses from RNA-sequence data. PACIFIC used *k-mers* representation for nucleotide sequences and assigns them to numerical tokens. A continuous vector space is used to convert tokens into dense representations. A deep learning-based tool, called autoBioSeqpy (Jing et al., 2020), was developed for the classification of biological sequences. Two sequence encoding methods (one-hot and dictionary-based) were used for bases/amino acids.

The aforementioned studies extracted important features, such as CpG-based features (Arslan, 2021b; Arslan & Arslan, 2021; Arslan, 2021a), representative genomic sequences (Lopez-Rincon et al., 2021), features extracted using the discrete Fourier transform, discrete Cosine transform and seven moment invariants (Naeem et al., 2021), intrinsic genomic signatures (Randhawa et al., 2020) ((sub)sequences of length 1 to 7 in conjunction with chaos game numerical representations), biomarkers (Singh et al., 2021), one-hot and dictionary encoding (Kuzmin et al., 2020; Jing et al., 2020), *k-mers* and kernel approximation-based features (Ali et al., 2021; Gunasekaran et al., 2021; Mateos et al., 2021), and intrinsic dinucleotide genomic signatures (Dlamini et al., 2021). Extracted features were then used for classification and detection purpose. Some studies only considered specific parts of the genome sequence, such as the Spike or Surface gene (Ali et al., 2021; Nawaz, Fournier-Viger, & He, 2022; Qiang et al., 2020) and some only used sequences that contain four bases. The PMBC (Pattern Mining from Biological sequences with wildcard Constraints) algorithm (X. Wu, Zhu, He, & Arslan, 2013) mines frequent patterns in biological sequences with a self-adaptive gap under the one-off condition. Besides random data, three human DNA sequences (AX829174, AY315625, and AY315623) obtained from NCBI were analyzed.

Differently from prior works, this study extracts sequential frequent patterns from whole genome sequences that can be used for reliable classification and detection purposes. More precisely, SPM algorithms are applied to the prepared corpus to find patterns (subsequences of bases) that appear frequently in genome sequences, and are

later used as features in the classification process. Moreover, the developed dataset of complete genome sequences contains bases other than A, C, G and T. However, in the results, we find that the inclusion of bases other than A, C, G and T plays no major role in classification/detection as they occur rarely in the sequences and thus are not present in frequent patterns.

3 SPM4GAC

The proposed SPM4GAC framework (Figure 1) to analyze and classify genome sequences consists of three main steps:

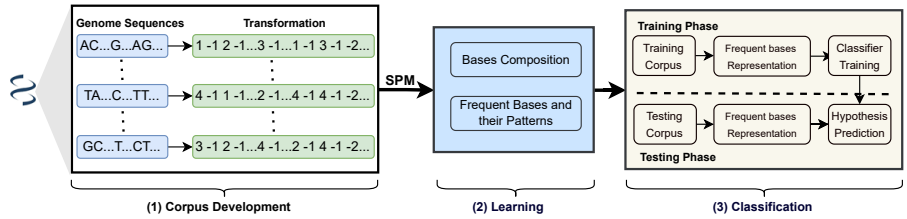

Fig. 1: SPM4GAC framework to analyze and classify genome sequences involves three steps: (1) Corpus development, (2) Learning via SPM, and (3) Classification via the discovered frequent sequential patterns of nucleotide bases in the created corpus by training various classifiers.

1. *Development of corpus*: First, genome sequences are obtained and converted into a discrete sequences corpus, where a distinct positive integer is used to encode each nucleotide.
2. *Using SPM for learning*: Second, frequent sequential patterns of nucleotide bases are discovered by invoking SPM algorithms on the transformed corpus.
3. *Using discovered frequent patterns in the classification*: Third, obtained frequent sequential patterns of nucleotide bases, discovered in step 2, are used to classify the genomes that belong to different virus families. The classification process contains two main tasks: (1) the training phase comprises two subphases, performed sequentially, representation of frequent nucleotides and training of a classifier. (2) the testing phase comprises three subphases, representation of frequent nucleotides, hypothesis prediction and evaluation. In the classification process, frequent sequential nucleotides are represented using both string-based and integer-based formats.

   The following subsections provides more detail for the three steps of SPM4GAC.

3.1 Corpus Development

The GenBank (Sayers et al., 2020) database is used to acquire the sequencing data of RNA viruses of various types: (1) positive-sense single-stranded (+ss), (2) negative-

sense single stranded (-ss), (3) negative-sense reverse transcriptase (-RT) and double-stranded (ds) (Table 1). Some genome sequences have limited research potential because they are smaller than the reference sequence or they contain a large number of ambiguous letters. We filter and select only those genomes that are complete and have a high coverage. Table 1 also provides details for the total bases (TB), minimum, maximum and average length of genome sequences in each virus family. We can see that on average, a genome sequence has an average length of 11,197. Genome data must first be converted into a suitable format that meets the following two primary conditions in order to be used with SPM:

- Obtained sequences must be transformed into sequences of discrete-type elements (items). This makes it possible to discover interesting hidden patterns in the data.
- To represent the data as discrete sequences, the set of items should be selected carefully. This requirement ensures a suitable abstraction that preserve all meaningful information and omit information that is redundant or irrelevant.

Table 1: Viruses genome sequences and their statistics. TB, MinL, MaxL and ASL represent total bases, minimum, maximum and average length of a sequence.

Virus	Samples	RNA Type	Redundant samples (%)	TB	MinL	MaxL	ASL
Dabie Banda	2,806	(+/-)ssRNA	29 (1.03)	10,394,666	1,674	6,386	3,704
Dengue	4,788	(+)ssRNA	285 (5.95)	50,250,495	1,485	11,195	10,495
Ebola	657	(-)ssRNA	57 (8.67)	12,423,106	18,277	19,897	18,908
Hanta	951	(-)ssRNA	47 (4.94)	3,542,253	204	6,761	3,724
Hepaci	1,206	(+)ssRNA	322 (26.69)	11,301,523	5,967	11,013	9,371
HIV	6,740	(-RT)ssRNA	1,124 (16.67)	59,807,993	1,103	10,514	8,873
Influenza	11,241	(-)ssRNA	271 (2.41)	18,542,776	246	2,867	1,649
Measles	759	(-)ssRNA	59 (7.77)	11,130,696	395	19,800	14,644
MERS	657	(+)ssRNA	105 (15.98)	19,737,610	23,327	30,484	30,042
Noro	1,130	(+)ssRNA	34 (3.00)	8,524,533	6,222	7,778	7,453
Rabies	2,422	(-)ssRNA	408 (16.84)	23,616,784	405	13,152	9,750
Rhino	885	(+)ssRNA	80 (9.03)	6,251,840	867	7,202	7,064
Rota	2,494	dsRNA	47 (1.88)	4,159,710	357	3,538	1,667
SARS-CoV-2	12,502	(+)ssRNA	5,158 (41.25)	372,155,530	29,490	29,903	29,767
West Nile	1,798	(+)ssRNA	116 (6.45)	19,507,357	8,916	11,355	10,849
Total	51,036		8,142 (15.95)	41,336,356	6,595	12,789	11,197

For transformation, the "*nucleotides to integers*" abstraction (Nawaz et al., 2021) is used. In such abstraction, each nucleotide is converted into a unique item, which is represented with a positive integer. This broad abstraction enables the use of different SPM algorithms.

Table 2: Nucleotide bases IUPAC codes.

Base (Code)	Base (Code)	Base (Code)
Adenine (A)	Cytosine (C)	Guanine (G)
Thymine (T)	A/G (R)	C/T(Y)
C/G (S)	A/T (W)	G/T (K)
A/C (M)	C/G/T (B)	A/G/T (D)
A/C/T (H)	A/C/G (V)	A/C/G/T (N)

The obtained genome sequences, downloaded as FASTA files, contain the information for the genome, followed by a nucleotides sequence. The four basic nucleotide bases are A (Adenine), C (Cytosine), G (Guanine) and T (Thymine) in DNA or U (Uracil) in RNA. However, there are some other nucleotides that represent different combinations of four bases (Table 2) (Johnson, 2010). For example, R (known as puRine) can be either A or G, and Y (known as pYrimidine) can be either C or T. Similarly, N can be any of the four bases. Here, we call them ambiguous or *redundant nucleotides* (RN) as they occur rarely. After the information field is removed, the whole genome sequence, shown as Ns, is a sequence of nucleotide bases. All of these nucleotides sequences are combined to create a corpus of discrete sequences. This corpus has the following formal definition.

Definition 1 (Nucleotide set) Assume that the set of all main nucleotides is formally described as $NB = \{A, C, G, T, R, Y, S, W, K, M, B, D, H, V, N\}$. The notation $|NB|$ refers to its cardinality and is equal to 15.

Definition 1 provides the following representation of a genomic sequence and a genome sequence corpus.

Definition 2 (Genome sequence) A list of nucleotides is called a genome sequence, $GS = \langle NB_1, NB_2, ..., NB_n \rangle$, such that $NB_i \in NB$ $(1 \leq i \leq n)$.

Definition 3 (Corpus of Genome sequences) A list of genome sequences creates a *genome sequences corpus* (GSC), which is defined as $GSC = \langle GS_1, GS_2, ..., GS_p \rangle$, where each genome has a unique identifier (ID). For instance, Table 3(a) shows a GSC that contains five genomes with IDs 1, 2, 3, 4 and 5. In this example and the rest of the paper, the commas between *NBs* are omitted to be consistent with the FASTA format of genome sequences.

Table 3: (a) A sample of *GSC* and (b) Bases as integers in genome sequences.

(a)

ID	Sequence
1	⟨{AATAACGG.....}⟩
2	⟨{TGCAATAG.....}⟩
3	⟨{CAGGTGTT.....}⟩
4	⟨{CCCTAATC.....}⟩
5	⟨{TGTAAACC.....}⟩

(b)

ID	Sequence
1	1 -1 1 -1 4 -1 1 -1 1 -1 2 -1 3 -1 3 -1 -2
2	4 -1 3 -1 2 -1 1 -1 1 -1 4 -1 1 -1 3 -1 -2
3	2 -1 1 -1 3 -1 3 -1 4 -1 3 -1 4 -1 4 -1 -2
4	2 -1 2 -1 2 -1 4 -1 1 -1 1 -1 4 -1 3 -1 -2
5	4 -1 3 -1 4 -1 1 -1 1 -1 1 -1 2 -1 2 -1 -2

To enable the use of SPM algorithms on the corpus, the genome sequences are converted into integer sequences in the last step. After this step, each row in the corpus contain a sequence of nucleotides that are substituted with positive integers. For example, the bases *A, C, G*, and *T* are changed to 1, 2, 3 and 4, respectively. Other bases *N, R, Y, K, M, S, W, B, D, H* and *V* are replaced with 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15 respectively. Additionally, for SPM algorithms, a separator characters, such as -1 is added between bases and -2 is put after the last element of a row, that is a genome sequence, to indicate that it has ended (Fournier-Viger et al., 2016). Table

3(b) shows the five sequences from Table 3(a) that have been transformed into integer sequences.

To be more precise about how the transformation process is conducted, a formal description of the transformation process is given next. Let there be a genome sequence corpus $GSC = \langle GS_1, GS_2, ..., GS_p \rangle$. The transformation process consists of transforming each genome sequence GS_i of the corpus GSC ($1 \leq i \leq p$). For a genome sequence $GS = \langle NB_1, NB_2, ..., NB_n \rangle$, the transformed genome sequence is denoted as GS' and is defined as $GS' = \langle f(NB_1), -1, f(NB_2), -1, ..., f(NB_n), -1, -2 \rangle$ where $f : NB \rightarrow \mathbb{N}$ is a function mapping a nucleotide to an integer. The transformation function f is defined as follows: $f(A) = 1$, $f(C) = 2$, $f(G) = 3$, $f(T) = 4$, $f(N) = 5$, $f(R) = 6$, $f(Y) = 7$, $f(K) = 8$, $f(M) = 9$, $f(S) = 10$, $f(W) = 11$, $f(B) = 12$, $f(D) = 13$, $f(H) = 14$, and $f(V) = 15$.

The transformation process yield datasets that can be used directly with most SPM algorithms to extract patterns. To allow reproducibility, interested readers can find the original datasets and their transformations at github.com/saqibdola/SPM4GAC.

3.2 Learning using SPM

For its application in genome sequence analysis, following is the definition of frequent SPM.

Definition 4 (Containment of a genome sequence) Let $GS_x = \langle x_1, x_2, ..., x_n \rangle$ and $GS_y = \langle y_1, y_2, ..., y_m \rangle$ represent two genome sequences. GS_x is contained (or present), in GS_y (denoted as $GS_x \sqsubseteq GS_y$), if and only if $1 \leq i_1 < i_2 < ... < i_n \leq m$, s.t. $x_1 = y_{i1}$, $x_2 = y_{i2}, ..., x_n = y_{im}$. GS_x is a *subsequence* of GS_y if GS_y contains GS_x.

Definition 5 (Support measure) For a genome sequence GS_x, the *support*, denoted as $sup(GS_x)$, in GSC refers to how many (sub)sequences contain GS_x. It is defined as: $sup(GS_x) = |\{GS | GS_x \sqsubseteq GS \wedge GS \in GSC\}|$.

Definition 6 (Task of frequent SPM in a GSC) For a GSC and a $minsup > 0$ (minimum support threshold given by a user), the goal of frequent SPM is to find every *frequent genome (sub)sequences*. It is said that GS, a genome subsequence, is frequent if $sup(GS) \geq minsup$.

$\langle TAA \rangle$ in Table 3 is present in three genome sequences 1, 4, and 5 respectively. Thus its support is 3.

Discovering frequent sequential patterns in GSC is difficult since the genome sequences could be lengthy and repetitive. There can be up to 2^{n-1} different subsequences in a genome sequence with n nucleotides. This means that it is not efficient to find sequential patterns by first determining the support of every possible subsequence using a naive counting algorithm. For that reason, in the last decade, a number of effective and novel algorithms were designed that employ different optimization techniques to discover the exact solution for the SPM problem without requiring an exhaustive search space exploration.

Algorithms in SPM establish a total order relation $\prec$ on items to expedite the process of identifying sequential patterns and prevent the discovery of identical sequences multiple times. In this study, the order $\prec$ is defined as the lexicographical order based on nucleotides from NB ($A \prec C \prec G \prec T ... \prec V$). In SPM, algorithms either use a breadth-first search (BFS) or a depth-first search (DFS) to explore and find frequent patterns. ***BFS***-based SPM algorithms first scan the dataset to identify sequential patterns that comprise a single item (1-sequences) and have a high occurrence frequency. They then generate 2-sequences by performing *s-extensions* and *i-extensions* of 1-sequences. In the same manner, 2-sequences are used to generate 3-sequences and so on. This procedure repeats till no more sequences can be produced. On the other hand, ***DFS***-based SPM algorithms begin with sequences that comprise single items and work recursively to create larger sequences by applying *i*-extensions and *s*-extensions to one of these sequences. If a pattern is not extendable, the algorithms backtrack and extend other patterns. For this study, the following defines *s*- and *i*-extensions, which take one or more k-sequences and produce a (k+1)-sequence from them. A genome sequence $GS_x = \langle x_1, x_2, ..., x_p \rangle$ is a ***prefix*** of another genome sequence $GS_y = \langle y_1, y_2, ..., y_q \rangle$, if $p < q$, $x_1 = y_1$, $x_2 = y_2$, ..., $x_{p-1} = y_{p-1}$, where x_p is equal to the first $|x_p|$ items of y_n in the $\prec$ order. GS_y is an *s-extension* of GS_x, for an item n, if $GS_y = \langle x_1, x_2, ..., x_p, \{n\} \rangle$. This means that GS_x is a prefix of GS_y and the item n follows after all the itemsets of GS_x. Here, *i-extension* is not defined as it does not apply in our case. It is noteworthy to mention that SPM can be applied in a broader scenario where concurrent items are permitted in a sequence, than what is described in this study. However, this general case is not discussed here as nucleotides are totally ordered in genome sequences.

SPM algorithms avoid searching the entire search space by using the Apriori property, which states that for GS_x and GS_y, if GS_x is a (sub)sequence of GS_y, then GS_y must have a support less than or equal than that of GS_x's support. This is explained with a simple example. Suppose that a sequence $\langle C \rangle$ is having a support of 4. Then, the (sub)sequence $\langle CG \rangle$'s support should be less than or equal to 4. This Apriori property helps in reducing the whole search space. All the extensions of rarely occurring sequences are also infrequent and thus they are not considered sequential patterns. For instance, for a $minsup$ of 5, it is not required to find $\langle C \rangle$'s extensions because all of them are infrequent. SPM algorithms either use a horizontal database format (HDF) or a vertical database format (VDF). In HDF, each entry represents a sequence. On the other hand, a VDF shows the itemsets where each item (nucleotide) is present in the sequence database. Table 3(a) shows a horizontal genome sequence database. SPM algorithms differs from each other in the following aspects:

1. Which strategy is employed- DFS or BFS,
2. Which kind of database representation (VDF or HDF) and internal data structures are employed,
3. How the support measure for patterns is calculated for finding those frequent patterns that meet the user-specified $minsup$ constraint.

CM-SPAM (Fournier-Viger, Gomariz, Campos, & Thomas, 2014) and TKS (Fournier-Viger, Gomariz, Gueniche, Mwamikazi, & Thomas, 2013) are a few of effective SPM algorithms. CM-SPAM (Fournier-Viger et al., 2014) uses the CMAP (Co-occurrence

MAP) data structure for reducing the search space and discovering sequential patterns. Information about the co-occurrences of an item is stored in a CMAP. However, in CM-SPAM, setting the $minsup$ threshold is not intuitive. There could be no patterns found with a high $minsup$ and many patterns found with a low $minsup$. TKS (Top-k Sequential), which is an extension of CM-SPAM, addresses this limitation by providing the users with the option to select how many patterns (k) to find in the corpus. TKS uses tailored strategies to reduce the whole search space. Both TKS and CM-SPAM use a vertical database representation.

3.3 Using Frequent Sequential Nucleotide Patterns for Classification

The third step of SPM4GAC is to classify genome sequences using the frequent patterns discovered with SPM algorithms.

Genome sequences are generally long, as shown in Table 1. Examining the GSC closely showed that nearly every sequence had the four bases (A, C, G and T) occurring hundred or even thousand times, sometimes consecutively. In genome sequences, frequent sequential patterns of bases can replace this bases repetition for improved classification results. In the Section 4, we found that using frequent patterns of bases enhanced the performance of classification. To be more precise, frequent sequential patterns of nucleotides are used to classify RNA virus families in the SPM4GAC framework.

Binary classification is used in SPM4GAC to classify/detect each virus type separately in GSC. The definition 7 of binary classification gives a label "virus name" to each genome sequence that belongs to a specific virus type and assigns "Others" to genome sequences that do not belong to that virus type.

Definition 7 Let V represents the set of virus types. GS, a genome sequence, is labelled in relation to v for a particular viral class $v \in V$ as:

$$GS_v = \begin{cases} v, & \text{if } GS \in v, \\ Others, & \text{otherwise} \end{cases} \tag{1}$$

Type labels belonging to v are labelled as v in Equation 1, whereas labels for other types are labelled as $Others$ in order to train a binary model. For instance, for the Influenza virus, Equation 1 assigns "*Influenza*" to genomes belonging to this virus and "*Others*" to genomes of other viruses.

In the multi-class (MC) classification setting, each genome in CGS is designated with its corresponding class name. Fifteen distinct virus types are considered in this study (Table 1). Thus, a model can be trained in MC for accurate labeling of genomes to their respective type.

Models performance, for both binary and MC classification, is evaluated and compared by using six metrics: accuracy (ACC), precision (P), recall (R), F1 score, false positive rate (FPR), and Matthews correlation coefficient (MCC). ACC in this study is calculated by dividing the total number of virus types by the proportion of

correctly categorized virus types. The definitions of the six measures are provided in the Appendix.

In total, ten models are used, including (1) SVM (Support Vector Machine), NB (Naive Bayes), (3) kNN (k-Nearest Neighbor), (4) J48 (Decision Tree), (5) K^* (KStar), (6) LR (Logistic Regression), (7) (RF) Random Forest, (8) (SGDT) Stochastic Gradient Descent Text, (9) MNBT (Multinomial Naive Bayes Text) and (10) ZeroR. Three classification models (MNBT, SGDT and and ZeroR) are string (text)-based, while the remaining are integer-based. String-based models use four strategies for tokenization: (1) Alphabetic Tokenizer (AT), (2) Character NGram Tokenizer (CNGT), (3) NGram Tokenizer (NGT) and (4) Word Tokenizer (WT). The following section discusses how those four tokenizers affect models' performance. The performance of each model is evaluated by using 10-fold cross validation.

4 Results

For the experiments, a computer with an eleventh generation Core i5 processor and 16 GB of RAM was used. The SPMF (Fournier-Viger et al., 2016) tool was used to analyze and find frequent sequential patterns in the corpus. Java built SPMF is open source and offers easy-to-use implementations for over 230 algorithms for pattern mining. Moreover, open-source WEKA (Frank, Hall, & Witten, 2016), also built in JAVA, was used for the models' training and testing on discovered frequent sequential patterns of nucleotides. WEKA was used because it is a cross-platform, offers various ML models and data preparation tools, meta-learners. In addition to a CLI (command line interface), it has an user-friendly GUI (graphical user interface). The results of analyzing the genome sequences of 15 virus families by using the SPM algorithms are discussed next.

4.1 Frequent Patterns

In a preliminary experiment, the developed corpus GSC was first analyzed to find the frequently occurring nucleotides and redundant nucleotides (*RN*) to compare their occurrence frequencies for different viruses, and see if insights could be obtained. For this purpose, the Apriori algorithm (Aggrawal & Srikant, 1994) was first applied, which is a popular and efficient algorithm for frequent itemset mining (FIM), that is for counting the occurrences of individual values or sets of values in data. The obtained frequencies are given in Table 4. The displayed values for *GC* and *AT/GC* contents were calculated from genome sequences using a Python script that is available at: github.com/saqibdola/SPM4GAC. From these results it is found that genome sequences that belong to various virus families are AT rich except for the Hepaci and West Nile viruses. This means that genome sequences that belong to Hepaci and West Nile contain high GC content as compared to genome sequences of other viruses.

Note that in the above preliminary experiment, the Apriori algorithm was used for counting the frequencies of bases because it is a popular algorithm, and it has many efficient implementations available in commercial or open-source data analysis

software but other more efficient algorithms could have been used to obtain the same information. The above analysis using Apriori allowed us to study the frequencies of bases. However, this analysis remains limited since Apriori does not consider the sequential ordering of bases and does not ensure that bases appear contiguously in a sequence. For this reason, only the frequencies of individual bases found by Apriori are displayed in Table 4 and frequent patterns containing multiple bases are not reported. In other words, the Apriori algorithm is unable to find sequential relationships among bases.

Table 4: Extracted frequent bases, GC and AT/GC ratio.

Bases	**Dengue**	**Ebola**	**MERS**	**HIV**	**Influenza**	**Rabies**	**Rota Virus**	**Rhino**
A	16,279,238	3,950,626	5,176,581	21,621,106	6,150,122	6,737,671	1,518,158	2,076,576
C	10,346,422	2,667,182	3,988,412	10,570,284	3,596,428	5,195,917	631,627	1,172,485
G	12,913,744	2,461,603	4,129,121	14,310,427	4,492,900	5,456,877	749,196	1,243,521
T	10,708,322	3,342,358	6,438,632	13,256,397	4,298,300	6,224,481	1,260,659	1,758,747
RN	2,769	1,337	4,864	49,779	5,026	1,838	70	511
GC	46.20	41.20	41.10	41.60	43.60	45.10	33.10	38.60
AT/GC	1.16	1.42	1.43	1.40	1.29	1.21	2.01	1.58
Bases	**SARS-CoV-2**	**West Nile**	**Noro**	**Dabie**	**Measles**	**Hanta**	**Hepaci**	
A	111,129,474	5,330,138	2,393,619	2,762,812	3,242,426	1,135,264	2,312,262	
C	68,136,870	4,345,791	2,124,232	2,293,859	2,657,697	613,041	3,328,455	
G	72,953,276	5,590,500	2,070,037	2,780,521	2,661,026	755,646	3,170,521	
T	119,477,777	4,224,812	1,935,632	2,557,414	2,597,064	1,037,857	2,481,019	
RN	2,529,366	16,116	1,115	60	22,483	445	9,266	
GC	37.95	50.90	49.20	48.80	47.40	38.60	57.50	
AT/GC	1.63	0.96	1.03	1.04	1.10	1.58	0.73	

Apriori limitations led to the development of efficient algorithms such as TKS and CM-SPAM. (Fournier-Viger et al., 2013, 2014, 2017; Nawaz, Fournier-Viger, Nawaz, Chen, & Wu, 2022). They are able to identify more meaningful information and patterns in the data. The *top-k* frequent sequential patterns of bases are found in the corpus by using TKS. In contrast to TKS, $minsup$ needs to be set by the user for running CM-SPAM. Table 5 lists a few frequent sequential patterns of bases, having different lengths, that the TKS and CM-SPAM identified in the genome sequences of 15 viruses. Table 5 offers some helpful insights regarding bases' frequent occurrences. We found that discovered frequent patterns in each virus family do not contain any RN as they occur rarely. Thus, the discovered frequent patterns only contain 4 bases (A, C, G and T). We observed that the process of pattern mining in genome sequences was fast. However, for a virus type that has long genome sequences, such as Ebola, MERS and SARS-CoV-2, we need to fine tune some parameters of both algorithms to find frequent patterns of bases. To gain more insights, the frequent patterns discovered in a raw RNA sequence are visualized in Figure 2. Each discovered pattern is a multiple of three nucleotides (codon) that encodes 20 different amino acids or stop signals. In the top of the figure, the raw RNA sequence is shown with the occurrences of each pattern displayed in with a different color. At the bottom left corner of the figure, the frequent sequential patterns are listed with their colors. It can be observed that some sequential patterns such as *AAAGAT* and *ATC* appear multiple times at different locations of the sequence. The combination of the patterns found in a sequence can be viewed as a description of the sequence.

Table 5: Extracted frequent sequential patterns of bases in each virus family.

	Dabie	Dengue	Ebola	Hanta	Hepaci
TKS	AAA	ATC	CCC	GCG	GGA
	CCA	ATT	TTA	TCG	TAT
	TGGATG	AAAGAT	CGTGGGGAA	ACTA	GCCT
	CCTGGA	GAAGGA	AGTCTA	ATGTG	TACA
	AAAAAGAAG	ATCCTGCTG	GGAAGATTA	ATTGA	CTTTC
	AAGACAATG	GAGACCCCT	ATGATTTTG	ATACT	ACGGC
	AAGACAATGATG	CAATATGCTGAA	ATGAAGATTAAG	AAAGAA	CTGGCT
	AAGACCCCCTTC	GAAGCTGTACGC	ATGAAGATTAAG	TGATGA	CTCCAA
	HIV	**Influenza**	**Measles**	**MERS**	**Noro**
	ACA	CGT	ACG	ACCA	GGT
	GAC	CGC	TTCC	CCAA	GTA
	GGGAGT	GCCG	GTGC	CATCC	CACT
	CACCAA	TGCCC	GTGCC	ATACTA	GGAC
	GATGGCAGG	ACTAC	GTGCGG	ATAGCA	GATTC
	ATGGCCAGT	AAATTG	AGACTGG	GCAAAGC	TCAGT
	GGTGCGAGAGCG	GAGAAC	CTGCTTGC	ACAAGCAG	ACATGA
	TGGAAAGGTGAA	ATGATG	AGATTCCTC	GTCAATATT	CACCAA
	Rabies	**Rhino**	**Rota**	**SARS-CoV-2**	**West Nile**
	CCA	TTA	AAA	CAAG	GCC
	ACGG	CTC	AAG	CTAAT	AAT
	AAGAT	GGGC	AAAT	CACTCA	AAAGAT
	TACTGC	TTAC	AAACA	CGTCTGC	CAGGCC
	CTATTC	AGATGA	AATGC	CAACAAG	AGGTCCTTC
	GACTATG	TGGACA	AATGAC	CACACTAAA	TGAGAATGG
	ATAGTGAA	GGTGGTGGA	AATTAG	AAGGGTGGT	CCTGGCTGTT
	TGATGTAT	TGGTGGTGG	AATTAG	CCGGAAGCCA	ACCTGGCTGTT
	TGTGAAAAAA	CGTGGCTGCCT	AAAAGAT		
CM-SPAM	**Dabie**	**Dengue**	**Ebola**	**Hanta**	**Hepaci**
	AAA	AAA	AAA	AAA	AAA
	AAC	AAC	AAC	AAC	AAC
	AAAATC	AAAC	AAAA	AACA	AAAG
	AAACCA	AAGA	AAAG	AACC	AAAC
	AAACTCCAC	AAGAG	AAACA	AACCT	ACCTA
	AAATGCTCC	AACTT	AAACC	AACTT	ACCAAG
	AACTCCACTGCA	AAGACA	AAATGG	AGCTCA	ACAGCT
	AAAAAGAAGACA	AAGAGA	AAATTC	AGGAAC	ACCTGGA
	HIV	**Influenza**	**Measles**	**MERS**	**Noro**
	AAA	AAA	AAA	AAA	AAA
	AAG	AAAT	AAAAG	AAAA	AAC
	AAAGGA	AAAGT	AAAAAAC	AAAAC	AAAG
	AAAACA	AAGTT	AAAACC	AAAAGA	AAAC
	AAAACAAAT	AATGAA	AAAACGT	AAAATGT	AAACG
	AAAAGCATT	ACCAAA	AAAACTTA	AAACAAC	ACATGA
	AAAGGGGGGATT	CAATTG	AAAACTTAG	AAACACTG	AAATTT
	AAATAAAATAGT	AGGACA	AAAAGAAACA	AAACACTGT	AAACTGA
	Rabies	**Rhino**	**Rota**	**SARS-CoV-2**	**West Nile**
	AAA	AAA	TCCG	AAA	AAA
	AACT	AAAG	GGCG	AAG	AAC
	AAAAC	AAACA	CCATC	AAAAAC	AACAAA
	AAACG	AACTCA	AGACAA	AAAAAG	AACAAC
	AAAGAG	AAGCACT	TGATAA	AAAAACC	AACACCTTC
	AAAGCGG	AATAAAT	AAGAAA	AAAAAGG	AAAACCATG
	AAAGGGCT	AATCAGA	TTTTAAA	AAAAAGGT	AAAACCATGGGA
	AACACTTCT	AATGTTGG	AGAAAAT	AAAAATTAT	AAAACAAAAGAA

4.2 Binary and MC Classification Results

For both binary and MC classification, the default hyperparameters for classifiers provided in WEKA version 3.8.6 were used. TKS and CM-SPAM are used to identify frequent 100, 200, 300 and 400 patterns of bases in each virus family. The discovered frequent patterns are preprocessed further to ensure that each pattern has 3 different bases, at least. The primary goal of finding patterns of various counts, such as 100 patterns, 200 patterns, 300 patterns and 400 patterns, is to investigate their effect on

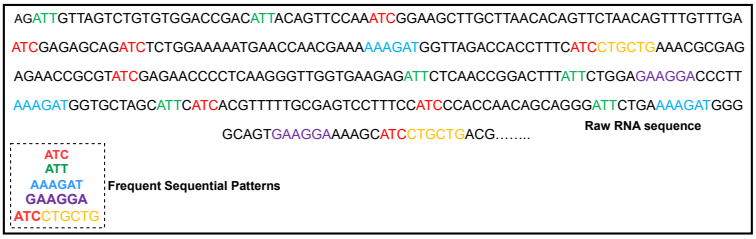

Fig. 2: A sample of frequent sequential patterns found in a raw RNA sequence.

the models' performance. Table 6 provides the classification results of models, both for binary and MC, when discovered frequent patterns are used as features.

In each table, the format ($\frac{100(200)}{300(400)}$) is used to provide the results for each model's metrics. Table 6 first entry, $\frac{27.26(27.16)}{27.57(27.83)}$, indicates that the NB classifier obtained an ACC of 27.26%, 27.16%, 27.57% and 27.83% with 100, 200, 300 and 400 patterns, respectively, that were discovered by using TKS. To keep the number of tables in this part to a minimum, metrics results are presented in the aforementioned format. Four tokenization algorithms are used with two string-based models (SGDT and MNBT): AT, CNGT, NGT and WT. For both MNBT and SGDT, CNGT outperformed AT, NGT and WT. The findings for MNBT and SGDT using the CNGT strategy are provided in Table 6. Whereas, ZeroR generated an ACC of 93.33 for all virus families. That is why they are not included in the tables. For binary classification, on overall SGDT using CNGT outperformed MNBT with CNGT. Training set results are obtained for MC classification. Moreover, SGDT in WEKA can only perform binary classification and not MC classification. In contrast to MNBT and ohter integer-based classifiers, SGDT was slow. On the other hand, RF was slow, followed by LR, for integer-based classifiers.

When it came to binary classification, all the classifiers performed better on CM-SPAM's patterns compared to TKS's patterns. For MC classification, kNN, K* and RF outperformed others on TKS's patterns as compared to CM-SPAM's patterns. For TKS's patterns, J48 and RF, which are tree-based models, performed better than LR, SVM, NB and kNN. On the other hand, J48 outperformed RF in binary classification. The opposite is true for MC classification, where RF outperformed J48. NB performed worst in all integer-based classifiers. Tree-based models performed better than others others as all the discovered sequential frequent patterns are utilized in the classification process where each frequent pattern contains nucleotide bases only, which are regarded as features.

We observed the same behavior for classifiers with CM-SPAM's patterns. J48 and RF outperformed other integer-based models and J48 performed better than RF for binary classification. For MC classification, we found that the classifiers NB, SVM, LR and MNBT performed better as compared to their performance on patterns discovered by using TKS. To avoid class imbalance, the frequent sequential patterns discovered in the first 657 genome sequences of each virus is used in the classification process. It is noteworthy to mention that every integer in the GSC is substituted with

Table 6: Classifiers accuracy on frequent sequential patterns of bases discovered in each virus family by using TKS and CM-SPAM.

Classifiers accuracy on frequent patterns of bases discovered by using TKS								
P	**Dabie**	**Dengue**	**Ebola**	**Hanta**	**Hepaci**	**HIV**	**Influenza**	**Measles**
NB	27.26(27.16)	90.33(93.33)	57(22.26)	62.66(61.70)	48.66(46.70)	90.86(92.96)	35.60(35)	89.53(89.40)
SVM	27.57(28.83)	92.08(92.15)	26.31(30.76)	60.95(60.07)	46.06(45.05)	93.28(93.31)	36.11(39.43)	88.88(89.06)
kNN	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)
K*	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)
J48	86(87.33)	86.40(87.70)	87.66(93.30)	87.20(87.23)	87.06(87.56)	87.66(88.50)	87.06(86.93)	89(90.10)
RF	88.17(87.38)	87.60(87.53)	88.46(88.70)	87.20(87.02)	87.17(87.83)	87.97(87.81)	86.28(86.80)	91.66(91.46)
LR	92.46(93.06)	92.13(92.70)	92.40(92.73)	92.66(93.23)	92.33(93.10)	93(92.80)	92.60(93.03)	93.20(93.60)
MNBT	93.28(93.31)	93.06(93.13)	92.93(92.90)	93.33(93.33)	93.26(93.30)	92.97(92.88)	93.31(93.33)	94.28(94.48)
SGDT	93.33(93.33)	93.33(93.13)	93.33(93.13)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	92.66(94.46)
	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	94.28(94.48)
	91.60(91.46)	92.06(91.63)	92.40(91.66)	91.20(90.16)	90.66(90.16)	92.46(92.30)	91.60(90.76)	93.46(93.96)
	91.42(91.60)	91.37(91.83)	91.97(91.96)	90.33(90.93)	90.68(91.31)	91.84(92.05)	90.55(91.20)	94.35(95.01)
	93.10(93.20)	93.06(93.33)	93.20(93.30)	93.13(93.10)	93.26(93.16)	93.06(93.20)	93.13(92.26)	93.13(93.23)
	93.26(93.26)	93.22(93.26)	93.28(93.25)	93.24(93.30)	93.16(93.28)	93.24(93.26)	93.31(93.30)	93.17(93.01)
	91.93(92.73)	91.13(91.10)	92.20(92.63)	88.40(92.36)	90.53(89.43)	88.60(89.20)	92.33(92.40)	84.26(83.30)
	92.24(91.13)	91(89.35)	92.04(92.96)	93.08(93)	90.42(87.33)	90.40(85.11)	92.77(91.75)	83.66(83.40)
	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.60(95.06)
	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	94.97(95.61)
P	**MERS**	**Noro**	**Rabies**	**Rhino**	**Rota**	**SARS-CoV-2**	**West Nile**	**MC**
NB	89.33(89)	42.66(37.66)	28.33(21.70)	30.26(29.03)	29.53(29.36)	93.60(93.96)	51.53(46.63)	17.86(17.30)
SVM	89.24(89.18)	35.95(36.48)	22.75(22.61)	28.68(28.93)	29.04(29.93)	93.37(93.48)	42.48(30.27)	16.35(16.20)
kNN	93.33(93.40)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.13(93.73)	93.33(93.33)	20.93(19.63)
K*	93.42(93.46)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.17(93.33)	93.33(93.31)	19.42(19.68)
J48	89.66(90.10)	85.73(88)	88(88.03)	87(86.96)	88.33(88)	92.46(94.23)	87.93(87.96)	88.26(79.10)
RF	89.97(90.86)	86.95(87.15)	87.84(87.86)	86.93(87.35)	87.17(87.18)	93.99(94.35)	88.37(88.46)	71.51(65.50)
LR	93.06(92.63)	92.40(93.23)	92.80(92.76)	92.46(93.23)	92.86(93.06)	95.20(95.86)	92.40(92.80)	88.20(79)
MNBT	93.04(90.86)	93.26(93.30)	92.97(93.13)	93.22(93.31)	93.24(93.33)	95.79(96.01)	92.80(93)	71.17(64.96)
J48	93(92.60)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	96.60(96.50)	93.33(93.33)	55.73(51.43)
RF	93.17(92.86)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	96.33(96.56)	93.33(93.31)	47.93(44.55)
LR	93(93.16)	91(90.80)	92.40(91.66)	91.46(91.06)	92.06(90.70)	96(96.73)	92.13(91.93)	88.26(79.10)
MNBT	92.80(93.05)	90.73(91.11)	91.20(91.70)	90.91(91.20)	90.93(91)	96.48(96.80)	91.75(91.80)	71.51(65.50)
SGDT	93.20(93.43)	93.33(93.23)	93.26(93.26)	93.20(93.30)	93.26(93.26)	94(94.63)	93.20(93.23)	21.86(20.30)
	93.37(93.40)	93.24(93.26)	93.26(93.21)	93.26(93.25)	93.20(93.25)	93.73(93.56)	93.24(93.25)	20.46(20.13)
	92.86(93)	91.53(92.26)	92.73(93.26)	91.93(92.96)	87.20(87.16)	92.53(91.76)	89.40(90.86)	22.13(20.33)
	92.93(92.43)	92.82(92.18)	93.17(92.96)	93.28(92.51)	87.46(86.63)	93.26(97.56)	90.02(86.18)	19.86(23.53)
	93(93.46)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	94.06(95.53)	93.33(93.33)	=(=)
	93.46(93.60)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	95.57(**98.86**)	93.33(93.33)	=(=)

Classifiers accuracy on frequent patterns of bases discovered by using CM-SPAM								
P	**Dabie**	**Dengue**	**Ebola**	**Hanta**	**Hepaci**	**HIV**	**Influenza**	**Measles**
NB	86(85.76)	59.20(47.20)	54(53.86)	49.05(53.83)	56(56.50)	89.53(86.63)	57.33(56.63)	94.53(94.63)
SVM	85.95(85.83)	42.33(41.78)	48.89(53.20)	51.88(53.93)	54.26(54.26)	85.66(85.20)	55.73(55.48)	92.55(93.26)
kNN	93.33(95.23)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	94(93.70)	95.46(95.33)	97.86(97.40)
K*	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	95(94.50)	95.57(95.53)
J48	94.93(95.80)	90.80(93.76)	89.46(90.03)	89.80(89.50)	90.53(90.43)	95.53(92.83)	93.53(92.50)	98.13(97.66)
RF	95.37(95.65)	91.71(91.31)	90.48(90.51)	90.40(89.96)	90.93(91.66)	92.33(92.76)	92.33(91.88)	97.11(97.53)
LR	97.26(97.50)	94(93.76)	93.20(93.36)	93.33(93.33)	93.53(93.23)	95.13(95.33)	93.93(94.03)	98.33(98.50)
MNBT	97.26(97.31)	93.88(93.66)	93.46(93.40)	93.26(93.33)	93.31(93.61)	95.33(95.43)	93.77(93.90)	98.04(98.48)
J48	96.33(96.86)	94.06(93.73)	93.33(93.43)	93.33(93.33)	93.26(93.33)	94.93(94.86)	94.93(95.03)	98.73(98.60)
RF	96.75(96.78)	94.26(93.83)	93.33(93.23)	93.26(93.13)	93.33(93.66)	94.60(94.93)	94.57(94.55)	97.84(98.48)
LR	97.06(97.26)	93.33(93.33)	92.20(91.76)	92.46(91.96)	92.46(92.50)	94.80(94.80)	94.53(93.96)	98.86(98.83)
MNBT	97.15(97.20)	92.22(92.90)	91.75(91.90)	92.17(92)	92.31(92.55)	94.62(94.76)	93.60(93.23)	98.33(98.60)
SGDT	93.33(95.16)	93.46(93.50)	93.33(93.23)	93.20(93.23)	93.33(93.30)	94.20(93.73)	95.26(95.26)	97.46(94.63)
	94.62(94.53)	93.28(93.33)	93.30(93.33)	93.28(93.33)	93.28(93.31)	93.31(93.18)	94.97(94.70)	95.73(95.63)
	94.13(94.70)	92.60(84.13)	92.53(86.20)	90.53(88.63)	85.86(85.13)	88.80(86.20)	86.20(85.06)	98.13(97.20)
	91.97(92.50)	91.82(92.51)	93.46(89.15)	93.17(93.28)	93.24(90)	88.64(86.80)	93.02(93.16)	96.28(96.06)
	96.86(97.16)	94.53(93.66)	93.06(93.26)	93.33(93.33)	93.33(93.36)	94.86(95)	94.80(95.03)	98.93(98.96)
	96.55(96.63)	93.91(93.40)	93.22(93.36)	93.33(93.33)	93.33(93.23)	95(94.56)	94.60(94.38)	98.40(98.83)
P	**MERS**	**Noro**	**Rabies**	**Rhino**	**Rota**	**SARS-CoV-2**	**West Nile**	**MC**
NB	84.60(84.46)	44.66(43.86)	60.26(44.70)	42.73(38.03)	46.40(45.26)	84.13(86.06)	59(50.33)	30.86(30.60)
SVM	85.80(85.03)	45.93(41.48)	43.86(38.95)	39.02(42.38)	46.71(47)	89.57(91.03)	53.46(44.73)	27.51(29.05)
kNN	93.53(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	95.40(94.43)	93.33(93.33)	38(38)
K*	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	93.33(93.33)	95.11(96.15)	93.33(93.33)	35.68(37.08)
J48	93.80(93.13)	90.60(90.70)	93.53(91.83)	90.73(91.56)	90.53(89.76)	95.06(95.90)	92.53(93.03)	55.20(55.86)
RF	93.11(93.31)	90.51(90.60)	92.15(91.01)	91.04(91.40)	89.95(89.65)	96.28(96.16)	92.82(92.73)	55.26(56.55)
LR	95.73(95.76)	93.33(93.33)	95.73(94.66)	93.33(93.53)	93.26(93.16)	96.73(97.66)	95.20(95.43)	55.13(55.66)
MNBT	95.80(95.63)	93.35(93.31)	94.31(94.40)	93.48(93.38)	93.24(93.26)	97.84(98.05)	95.08(95.31)	55.04(56.28)
SGDT	95.20(95.50)	93.33(93.33)	94.80(94.16)	93.33(98.83)	93.33(93.46)	96.40(97.90)	94.86(95.66)	51.40(50.83)
	95.64(95.45)	93.33(93.40)	94.06(93.81)	93.28(93.76)	93.33(93.35)	98.33(98.61)	94.80(94.91)	50.71(51.06)
	95.53(95.26)	92.26(92.60)	95.60(93.60)	92(92.93)	91.93(91.70)	97.20(97.56)	94.33(95.06)	55.20(55.86)
	95.06(95.13)	92.46(92.01)	93.50(93.71)	92.51(92.75)	91.37(91.56)	98.06(98.03)	94.44(94.65)	55.26(56.55)
	94.46(93.70)	93.26(93.33)	93.20(93.30)	93.33(93.13)	93.26(93.26)	95.26(94.63)	93.53(93.30)	37.13(38.03)
	93.42(93.40)	93.33(93.30)	93.26(93.30)	93.33(93.33)	93.15(92.95)	95.22(96.15)	93.33(93.33)	35.17(35.90)
	92.06(95.06)	90.46(79.96)	87.46(84.16)	81.66(89.16)	86.73(84.46)	97.26(98.23)	85.80(99.93)	36.86(39.50)
	90.07(90.90)	93.33(91.20)	90.66(87.38)	90.08(87.88)	90.46(87.63)	90.13(90.23)	84.75(85.33)	28.04(27.73)
	95.73(95.80)	93.33(93.31)	94.46(94.76)	93(93.33)	93.33(93.20)	97.86(98.53)	94.60(**100**)	=(=)
	94.84(94.96)	93.33(93.33)	94.04(93.86)	93.33(93.33)	93.33(93.33)	94.62(94.83)	94(93.56)	=(=)

its corresponding nucleotide letter when using string-based models. This is done to see if string-based models outperform integer-based classifiers. We found that for binary classification, SGDT outperformed integer-based models.

For patterns discovered with both SPM algorithms, all classifiers performed better, overall, on all pattern lengths (100, 200, 300 and 400). In some cases of CM-SPAM patterns, in comparison to 200, 300, and 400 patterns, the models performed better on 100 patterns. Overall, J48 outperformed other integer-based models on patterns discovered with both SPM algorithms, followed by RF for binary classification. Whereas SGDT performed better than J48. The complete results of SGDT and J48 on CM-SPAM's patterns are provided in Table 7, where the value ? is displayed when no result can be computed by WEKA. It should be noted that the high MCC values for SGDT and J48 indicate that both classifiers were successful in correctly predicting

in the majority of the confusion matrix's four categories (TP, TN, FP and FN). The confusion matrix information for MC classification results obtained with kNN, that is similar to RF, is shown in Figure 3.

Table 7: SGDT and J48 results on patterns discovered by using CM-SPAM

SGDT results								
P	**Dabie**	**Dengue**	**Ebola**	**Hanta**	**Hepaci**	**HIV**	**Influenza**	**Measles**
ACC	96.86(97.16)	94.53(93.66)	93.06(93.26)	93.33(93.33)	93.33(93.36)	94.86(95)	94.80(95.03)	98.93(98.96)
FPR	96.55(96.63)	93.91(93.40)	93.22(93.36)	93.33(93.33)	93.33(93.23)	95(94.56)	94.60(94.38)	98.40(98.83)
P	0.38(0.34)	0.76(0.87)	0.91(0.91)	0.93(0.93)	0.92(0.91)	0.64(0.63)	0.56(0.60)	0.12(0.11)
R	0.43(0.41)	0.84(0.90)	0.93(0.92)	0.93(0.93)	0.93(0.92)	0.64(0.74)	0.58(0.65)	0.15(0.13)
F1	0.96(0.97)	0.94(0.92)	0.88(0.90)	?(?)	0.90(0.91)	0.94(0.94)	0.94(0.94)	0.98(0.98)
MCC	0.96(0.96)	0.94(0.91)	0.87(0.91)	?(?)	?(0.88)	0.94(0.94)	0.93(0.93)	0.98(0.98)
	0.96(0.97)	0.94(0.93)	0.93(0.93)	0.93(0.93)	0.93(0.93)	0.93(0.95)	0.94(0.95)	0.98(0.99)
	0.96(0.96)	0.93(0.93)	0.93(0.93)	0.93(0.93)	0.93(0.93)	0.95(0.94)	0.94(0.94)	0.98(0.98)
	0.96(0.96)	0.92(0.91)	0.90(0.90)	?(?)	0.90(0.90)	0.93(0.94)	0.94(0.94)	0.98(0.98)
	0.96(0.96)	0.91(0.90)	0.90(0.90)	?(?)	?(0.90)	0.93(0.92)	0.93(0.93)	0.98(0.98)
	0.71(0.74)	0.41(0.21)	0.05(0.08)	?(?)	0.06(0.10)	0.47(0.49)	0.50(0.50)	0.91(0.91)
	0.68(0.69)	0.28(0.12)	−.009(0.07)	?(?)	?(0.03)	0.49(0.41)	0.47(0.42)	0.86(0.90)
P	**MERS**	**Noro**	**Rabies**	**Rhino**	**Rota**	**SARS-CoV-2**	**West Nile**	
ACC	95.73(95.80)	93.33(93.31)	94.46(94.76)	93(93.33)	93.33(93.20)	97.86(98.53)	94.60(**100**)	
FPR	94.84(94.96)	93.33(93.33)	94.04(93.86)	93.33(93.33)	93.33(93.33)	94.62(94.83)	94(93.56)	
P	0.51(0.52)	0.93(0.93)	0.44(0.64)	0.91(0.93)	0.93(0.92)	0.27(0.18)	0.70(**0**)	
R	0.66(0.66)	0.93(0.93)	0.78(0.85)	0.93(0.93)	0.93(0.93)	0.61(0.62)	0.79(0.88)	
F1	0.95(0.95)	?(?)	0.96(0.94)	0.88(?)	?(0.88)	0.97(0.98)	0.93(**1**)	
MCC	0.94(0.94)	?(?)	0.93(0.93)	?(?)	?(?)	0.93(0.94)	0.93(0.91)	
	0.95(0.95)	0.93(0.93)	0.96(0.94)	0.93(0.93)	0.93(0.93)	0.97(0.98)	0.94(**1**)	
	0.94(0.95)	0.93(0.93)	0.94(0.93)	0.93(0.93)	0.93(0.93)	0.94(0.94)	0.94(0.93)	
	0.95(0.95)	?(?)	0.96(0.93)	0.90(?)	?(0.90)	0.97(0.98)	0.93(**1**)	
	0.93(0.93)	?(?)	0.92(0.91)	?(?)	?(?)	0.93(0.93)	0.92(0.91)	
	0.59(0.59)	?(?)	0.67(0.46)	0.04(?)	?(0.01)	0.81(0.87)	0.43(**1**)	
	0.46(0.48)	?(?)	0.32(0.27)	?(?)	?(?)	0.46(0.48)	0.31(0.20)	

J48 results								
P	**Dabie**	**Dengue**	**Ebola**	**Hanta**	**Hepaci**	**HIV**	**Influenza**	**Measles**
ACC	96.33(96.86)	94.06(93.73)	93.33(93.43)	93.33(93.33)	93.26(93.33)	94.93(94.86)	94.93(95.03)	**98.73**(98.60)
FPR	96.75(96.78)	94.26(93.83)	93.33(93.23)	93.26(93.13)	93.33(93.66)	94.60(94.93)	94.57(94.55)	97.84(98.48)
P	0.45(0.40)	0.77(0.82)	0.93(0.91)	0.93(0.93)	0.93(0.93)	0.70(0.63)	0.61(0.54)	0.15(**0.15**)
R	0.42(0.41)	0.75(0.85)	0.93(0.91)	0.93(0.89)	0.93(0.85)	0.70(0.64)	0.60(0.65)	0.26(0.19)
F1	0.96(0.96)	0.93(0.92)	?(0.92)	?(?)	0.87(?)	0.95(0.94)	0.94(0.94)	**0.98**(0.98)
MCC	0.96(0.96)	0.93(0.93)	?(0.89)	0.88(0.89)	?(0.92)	0.94(0.94)	0.93(0.93)	0.97(0.98)
	0.96(0.96)	0.94(0.93)	0.93(0.93)	0.93(0.93)	0.93(0.93)	0.94(0.94)	0.94(0.950)	**0.98**(0.98)
	0.96(0.96)	0.94(0.93)	0.93(0.93)	0.93(0.93)	0.93(0.93)	0.94(0.94)	0.94(0.94)	0.97(0.98)
	0.95(0.96)	0.92(0.91)	?(0.90)	?(?)	0.90(?)	0.93(0.93)	0.94(0.94)	**0.98**(0.98)
	0.96(0.96)	0.92(0.91)	?(0.90)	0.90(0.90)	?(0.91)	0.93(0.93)	0.93(0.93)	0.97(0.98)
	0.66(0.71)	0.33(0.26)	?(0.12)	?(?)	−.007(?)	0.47(0.48)	0.49(0.52)	**0.89**(0.88)
	0.70(0.70)	0.37(0.26)	?(0.07)	0.01(0.10)	?(0.23)	0.42(0.48)	0.46(0.44)	0.813(0.872)
P	**MERS**	**Noro**	**Rabies**	**Rhino**	**Rota**	**SARS-CoV-2**	**West Nile**	**MC**
ACC	95.20(95.50)	93.33(93.33)	94.80(94.16)	93.33(98.83)	93.33(93.46)	96.40(97.90)	94.86(95.66)	51.40(50.83)
FPR	95.64(95.45)	93.33(93.40)	94.06(93.81)	93.28(93.76)	93.33(93.35)	98.33(98.61)	94.80(94.91)	50.71(51.06)
P	0.58(0.52)	0.93(0.93)	0.60(0.79)	0.91(0.82)	0.93(0.91)	0.35(0.18)	0.67(0.57)	0.03(0.03)
R	0.50(0.53)	0.93(0.0.89)	0.79(0.81)	0.93(0.82)	0.93(0.75)	0.15(0.14)	0.66(0.64)	0.03(0.03)
F1	0.94(0.95)	?(?)	0.94(0.93)	0.90(0.92)	?(0.93)	0.96(0.97)	0.94(0.95)	0.68(0.62)
MCC	0.95(0.94)	?(0.91)	0.93(0.92)	0.88(0.92)	?(0.91)	0.98(0.98)	0.94(0.94)	0.57(0.62)
	0.95(0.95)	0.93(0.93)	0.94(0.94)	0.93(0.93)	0.93(0.93)	0.96(0.97)	0.94(0.95)	0.51(0.50)
	0.95(0.95)	0.93(0.93)	0.94(0.93)	0.93(0.93)	0.93(0.93)	0.98(0.98)	0.94(0.94)	0.50(0.51)
	0.94(0.94)	?(?)	0.93(0.92)	0.90(0.91)	?(0.90)	0.96(0.97)	0.93(0.94)	0.52(0.50)
	0.95(0.94)	?(0.90)	0.92(0.91)	0.90(0.91)	?(0.91)	0.98(0.98)	0.93(0.93)	0.51(0.51)
	0.52(0.56)	?(?)	0.48(0.34)	0.09(0.27)	?(0.13)	0.68(0.82)	0.47(0.57)	0.52(0.50)
	0.58(0.56)	?(0.14)	0.32(0.28)	0.02(0.26)	?(0.27)	0.86(0.88)	0.46(0.48)	0.49(0.50)

To determine whether RF significantly outperforms J48 and the other five integer-based models on patterns found by using CM-SPAM, a paired t-test is run in WEKA. The comparative results for the ACC of models are shown in Table 8. Bold entries indicate models that considerably under-performed RF, whereas blue-colored entries indicate those that significantly outperformed RF. NB and kNN performed significantly worst than RF. Whereas K* has almost the same performance as RF, indicating that in many cases, the performance gap between these two classifiers is not that significant. J48 outperformed, significantly, RF in most of the cases. T-test is also used to determine that in two models (SGDT and MNBT), which one performs significantly better than the other one. When using the CNGT, SGDT outperformed MNBT significantly.

The main findings are: (1) For binary classification: (a) classification models performed better on CM-SPAM's patterns, and (b) String-based model (SGDT) outperformed integer-based models. (2) Overall, J48 and RF outperfomed other models (NB, SVM, kNN K* and LR). (3) It is not possible to recommend a particular sequence pattern mining algorithm for MC classification as some classifiers performed better than others on TKS's patterns, while others performed better on CM-SPAM's

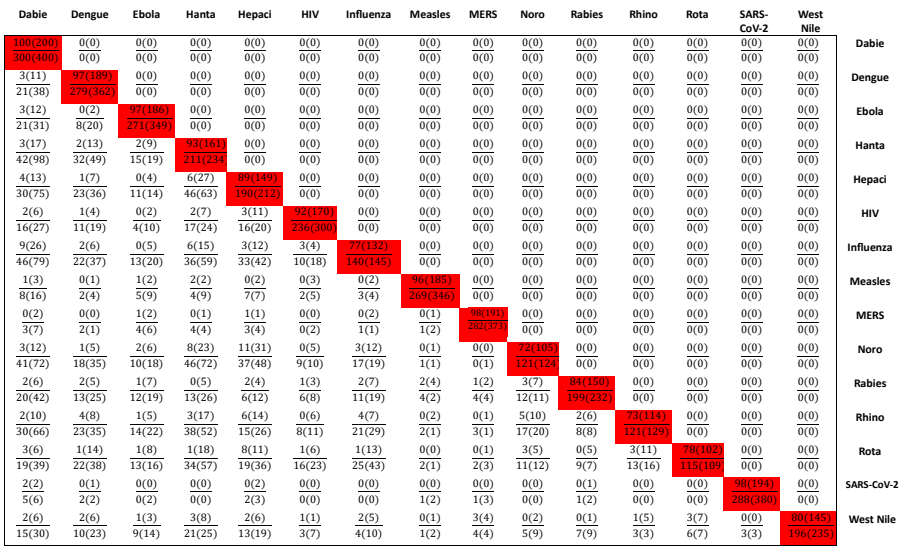

Fig. 3: MC classification confusion matrix for kNN on frequent patterns of bases discovered by using TKS.

patterns. (4) There was no noticeable difference in the performance of classifiers with variable pattern lengths. For CM-SPAM's patterns, the models' performance was better, in some cases, with 100 patterns than with 200, 300 or 400 patterns. (5) Rather than using the whole genome sequences, frequent sequential patterns of bases can be utilized efficiently in the classification/detection process. Table 1 shows that the average genome sequence of an RNA virus has thousands of nucleotides. On the other hand, the CM-SPAM's patterns only contain a maximum of 37 bases (Table 9).

4.2.1 Comparison

SPM4GAC is compared in this section with SOTA approaches for genome detection/classification, published in last three years.

Table 10 compares the performance of SPMG4GAC with previous approaches. SGDT (indicated in bold) of SPM4GAC achieved improved performance compared to kNN (Alshayeji et al., 2023; Arslan & Arslan, 2021; Arslan, 2021a), RF (Arslan, 2021b; Singh et al., 2021), SVM (Ahmed & Jeon, 2022), CNN (Lopez-Rincon et al., 2021; El-Dosuky et al., 2021; Gunasekaran et al., 2021) and PACIFIC (Mateos et al., 2021) for binary classification. SPM4GAC(J48) performed better than RF (Arslan, 2021b; Singh et al., 2021), SVM (Ahmed & Jeon, 2022) and CNN (Gunasekaran et al., 2021), and similar to kNN (Arslan & Arslan, 2021; Alshayeji et al., 2023) and CNN (Lopez-Rincon et al., 2021). (Alshayeji et al., 2023) used their method on human DNA sequences and used six classifiers in which kNN performed best with accuracy of 98.6%, for the prediction of human DNA sequence contigs. Their

Table 8: Paired t-test results for classifiers.

Dataset	RF	NB	SVM	kNN	K*
Dabie	97.04(97.32)	**86.05(85.69)**	**93.33(95.21)**	**95.36(95.75)**	97.18(97.53)
Dengue	97.11(97.22)	**85.85(85.79)**	**93.36(93.33)**	**95.57(95.73)**	97.27(97.34)
Ebola	93.49(93.20)	**59.11(47.60)**	93.33(93.33)	**91.15(91.48)**	94.11(93.77)
Hanta	93.18(92.86)	**42.27(41.72)**	93.33(93.33)	**91.67(91.23)**	93.87(93.71)
Hepaci	91.62(91.89)	**54.09(54)**	93.33(93.33)	**89.76(90.14)**	93.06(93.30)
HIV	91.96(91.84)	**48.88(53.22)**	93.33(93.33)	**90.41(90.38)**	93.45(93.39)
Influenza	92.26(91.80)	**49.35(53.82)**	93.33(93.33)	**90(89.65)**	93.33(93.33)
Measles	92.26(91.99)	**51.81(53.95)**	93.33(93.33)	**90.43(90.05)**	93.29(93.32)
MERS	92.58(92.45)	**55.85(56.53)**	93.33(93.33)	**90.97(90.76)**	93.58(93.19)
Noro	92.35(92.57)	**54.27(54.06)**	93.33(93.33)	**90.76(91.53)**	93.30(93.60)
Rabies	94.75(94.84)	**89.30(86.52)**	94(**93.66**)	**92.69(92.69)**	95.21(95.43)
Rhino	94.62(94.80)	**85.82(85.08)**	**93.33(93.33)**	**92.56(92.89)**	95.42(95.43)
Rota	94.37(94.14)	**57.33(56.62)**	95.45(95.33)	**93.43(92.79)**	93.99(94.06)
SARS-CoV-2	93.54(93.08)	**55.71(55.50)**	94.99(94.50)	**92.30(91.72)**	93.73(93.72)
West Nile	98.76(98.73)	**94.36(94.59)**	**97.99(97.29)**	**98.01(97.74)**	98.44(98.51)
MC	98.29(98.56)	**92.57(93.24)**	**95.60(95.58)**	**97.14(97.49)**	98.08(98.48)
	95.38(95.31)	**84.66(84.34)**	**93.5(93.33)**	**93.84(93.31)**	95.75(95.79)
	95.15(95.03)	**85.73(85.10)**	**93.33(93.33)**	**93.18(92.98)**	95.77(95.64)
	92.17(92.44)	**44.73(43.78)**	93.33(93.33)	**90.31(90.26)**	93.33(93.34)
	92.36(92.11)	**45.90(41.48)**	93.33(93.33)	**90.63(90.64)**	93.35(93.33)
	95.31(93.79)	**60.29(44.81)**	**93.33**(93.33)	**93.32(92.10)**	95.63(94.70)
	93.50(93.70)	**43.81(39)**	93.33(**93.33**)	**92.19(92.18)**	94.45(94.43)
	92.28(92.91)	**42.70(38.02)**	93.33(93.33)	**90.87(91.37)**	93.26(93.51)
	92.59(92.60)	**38.93(42.25)**	93.33(93.33)	**91.24(91.25)**	93.50(93.38)
	91.70(91.65)	**46.21(45.28)**	93.33(93.33)	**90.07(89.84)**	93.31(93.11)
	91.29(91.57)	**46.83(47.03)**	93.33(93.33)	**89.66(89.73)**	93.24(93.27)
	97.74(97.52)	**84.33(86.12)**	**95.34(94.59)**	**95.25(95.66)**	97(97.46)
	98.06(98.03)	**89.57(91.03)**	**95.17(96.17)**	**96.16(96.19)**	97.86(97.49)
	94.55(95.20)	**58.81(50.36)**	93.33(**93.33**)	**92.38(93.10)**	95.13(95.45)
	94.52(94.77)	**53.52(44.80)**	**93.33(93.33)**	**92.96(92.58)**	95.07(95.35)
	33.68(34.63)	**29.09(29.77)**	34.33(36.36)	**29.23(29.88)**	**32.56**(34.10)
	33.11(34.24)	**27.08(28.86)**	33.97(35.71)	**28.69(29.62)**	33.08(34.61)
Dataset	J48	LR	MNBT	SGDT	
Dabie	**96.41(96.88)**	**93.87(95.16)**	**96.79(94.69)**	96.79(97.22)	
Dengue	96.82(**96.71**)	**94.55(94.49)**	**92.06(92.53)**	96.47(96.59)	
Ebola	94.39(93.74)	93.50(93.48)	**91.49(83.47)**	94.53(93.59)	
Hanta	94.24(93.71)	93.30(93.33)	**91.80(92.50)**	93.87(93.44)	
Hepaci	93.33(93.34)	93.32(93.28)	**91.71(86.02)**	93.13(93.29)	
HIV	93.33(93.27)	93.31(93.31)	93.18(**88.90**)	93.24(93.35)	
Influenza	93.33(93.34)	93.24(93.28)	**88.87(87.82)**	93.31(93.33)	
Measles	93.28(93.26)	93.29(93.31)	**92.92**(93.22)	93.33(93.33)	
MERS	93.28(93.30)	93.25(93.28)	**83.57(84.98)**	93.43(93.25)	
Noro	93.31(93.67)	93.27(93.29)	**93.08(89.72)**	93.33(93.25)	
Rabies	95.06(95.04)	93.94(**93.79**)	**88.43(85.70)**	95.04(94.93)	
Rhino	94.64(94.80)	**93.34**(93.21)	**88.74(86.58)**	94.97(94.51)	
Rota	94.89(95.19)	95.31(95.35)	**83.41(84.67)**	94.49(95.12)	
SARS-CoV-2	94.59(94.60)	94.98(94.74)	**92.68(93)**	94.54(94.43)	
West Nile	98.65(98.66)	**97.61(97.06)**	**97.75(97.10)**	98.91(99.03)	
MC	97.98(98.48)	**95.67(95.64)**	**96.25(95.92)**	98.54(98.81)	
	95.19(95.64)	**94.45(93.63)**	**91.38(94.74)**	95.67(95.70)	
	95.65(95.41)	**93.45(93.44)**	**90.68(90.79)**	94.93(94.92)	
	93.33(93.28)	93.24(93.28)	**89.18(78.63)**	93.33(93.36)	
	93.33(93.40)	93.32(93.32)	93.31(**91.09**)	93.33(93.33)	
	94.93(94)	**93.28**(93.31)	**87.13(84.57)**	96.43(94.82)	
	93.97(93.94)	93.32(93.33)	**90.72(87.37)**	94.13(93.82)	
	93.25(93.82)	93.31(93.33)	**80.01(88.76)**	92.97(93.33)	
	93.30(93.68)	93.32(93.32)	**89.91(87.67)**	93.33(93.33)	
	93.32(93.31)	93.24(93.22)	**84.03(85.92)**	93.29(93.21)	
	93.33(93.39)	93.21(92.97)	**90.32(87.60)**	93.22(93.29)	
	96.67(97.78)	**95.32(94.62)**	97.31(98.23)	97.79(98.47)	
	98.40(98.50)	**95.30(96.20)**	**89.94(90.15)**	94.68(94.74)	
	95.01(95.56)	**93.48(93.31)**	**84.55**(99.93)	94.70(100)	
	94.73(94.92)	**93.31(93.33)**	**84.39(85.34)**	93.96(93.59)	
	40.11(42.94)	32.53(35.15)	−(−)	−(−)	
	41.65(43.09)	32.86(34.15)	−(−)	−(−)	

Table 9: Statistics for the CM-SPAM's patterns discovered in virus families.

Virus	**ASL**	**MaxL**	**Virus**	**ASL**	**MaxL**	**Virus**	**ASL**	**MaxL**
Dabie	10.85(12.68)	22(32)	MERS	11.04(10.86)	24(28)	Measles	19.02(19.18)	37(37)
	11.87(12.20)	32(32)		11.03(10.84)	33(33)		17.92(18.31)	37(37)
Dengue	7.96(7.28)	22(22)	Noro	5.86(5.88)	7(9)	Influenza	5.08(4.97)	6(6)
	7.52(7.28)	22(22)		5.85(5.86)	9(9)		4.93(4.92)	6(6)
Ebola	7.51(7.49)	12(12)	Rabies	9.68(8.51)	19(19)	West Nile	9.15(9.38)	21(21)
	7.33(7.30)	12(12)		8.02(7.89)	19(19)		8.87(9.13)	21(21)
Hanta	5.56(5.53)	7(7)	Rhino	5.44(5.40)	7(7)	HIV	10.89(10.52)	31(31)
	5.42(5.41)	7(7)		5.23(5.23)	7(7)		10.56(10.19)	31(31)
Hepaci	5.62(5.59)	7(7)	Rota	5.44(5.40)	7(7)	SARS-CoV-2	11.08(11.31)	15(15)
	5.58(5.59)	7(7)		5.23(5.23)	7(7)		11.50(11.49)	15(15)

method first found *k-mers* and the bag-of-words technique was then used for feature extraction. Extracted features were then fed into classifiers.

RF in (Arslan, 2021b) achieved the highest accuracy of 93% by using CpG based features in genome sequences of viruses. (Arslan & Arslan, 2021) achieved the highest accuracy of 98.4% when any of the six metric (Canberra, Chebyshev, Manhattan,

Kulezynski, Sorensen and Mean character) were used as a distance measure in kNN. Highest accuracy of 99.8% was obtained in (Arslan, 2021a) with kNN by combining similarity features features with CG-based features. RF in (Singh et al., 2021) achieved highest accuracy of 97.47% on derived biomarkers, in genomes, based on three-based periodicity properties. CNN in (Gunasekaran et al., 2021) achieved highest accuracy of 93.16% by using label and *k-mer* encoding for genome sequences of viruses. PACIFIC (Mateos et al., 2021), that performed embedding of *k-mer* and CNN filtering to BiLSTM layers, achieved average accuracy of 99.95% for each of the five virus classes.

Table 10: Comparison of SPM4GAC with recent classification/detection methods for genome sequences

Type	Best Models	ACC	FPR	R	P	F1	MCC
Binary	kNN (Alshayeji et al., 2023)	0.98	–	0.98	0.98	0.98	0.89
	RF (Arslan, 2021b)	0.93	–	0.93	0.93	0.93	–
	kNN (Arslan & Arslan, 2021)	0.98	–	0.99	0.98	0.98	–
	kNN (Arslan, 2021a)	0.99	–	0.99	0.99	0.99	0.99
	CNN (Lopez-Rincon et al., 2021)	0.98	–	–	–	–	–
	SVM (Ahmed & Jeon, 2022)	0.97	–	0.77	0.97	0.97	–
	RF (Singh et al., 2021)	0.97	–	0.96	–	–	–
	CNN (El-Dosuky et al., 2021)	0.99	–	0.99	0.99	–	–
	CNN (Gunasekaran et al., 2021)	0.93	–	0.98	0.90	0.94	–
	PACIFIC (Mateos et al., 2021)	0.99	0.003	0.99	0.99	–	–
	SPM4GAC(SGDT)	1	0	1	1	1	1
	SPM4GAC(J48)	0.98	0.150	0.98	0.98	0.98	0.89
MC	**SPM4GAC(RF)**	0.88	0.008	0.88	0.88	0.88	0.87

Although the studies (Randhawa et al., 2020; Naeem et al., 2021) produced classification results with 100% accuracy, Table 10 does not include their findings since the datasets they used had much fewer genome sequences. The binary classification results of the studies (Ali et al., 2021; Kuzmin et al., 2020; Nawaz, Fournier-Viger, & He, 2022; Nawaz, Fournier-Viger, He, & Zhang, 2023; Qiang et al., 2020) are also not included in Table 10 because only sequences of the Spike protein of viruses are considered for the classification process. Note that SPM4GAC(SGDT) performed better than SVM (Ali et al., 2021), SVM and DT (Kuzmin et al., 2020), SGDT (Nawaz, Fournier-Viger, & He, 2022) and RF (Nawaz, Fournier-Viger, He, & Zhang, 2023; Qiang et al., 2020). Majority of the listed methods only performed binary classification. We also include MC classification results for SPM4GAC(RF). Obtained results show that preforming classification on a reduced feature set obtained from frequent sequential patterns yield better performance, while reducing memory and computational requirements.

5 Conclusion

A framework (called SPM4GAC) was presented in this paper to analyze and classify genome sequences of various RNA viruses. A corpus that contains genome se-

quences of RNA viruses was first developed and formatted appropriately. The transformed corpus was then subjected to SPM algorithms to discover frequently occurring nucleotides and their frequent sequential patterns. Discovered frequent patterns of bases were then used in classification. Ten classification models, in which seven were integer-based and three were string-based, were used. The performance of the models were assessed by using six evaluation metrics. Results obtained for binary classification indicate that one string-based classifier, SGDT, and two integer-based classifiers, J48 and RF outperformed others. On the other hand, two integer-based classifiers, RF and kNN, outperformed others in MC classification. The main take away from obtained results is that shorter (or limited) genomes of RNA viruses, containing frequent occurring bases only, can be utilized for reliable prediction/ classification rather than whole sequences. SPM4GAC outperformed SOTA methods for genome sequence classification/detection and offers many interesting future work opportunities, such as:.

- Investigating the performance of SPM4GAC on DNA viruses that contain larger genomes as compared to RNA viruses.
- SPM4GAC is developed for genome sequences in *nucleotide* form. Extending this framework to analyze and classify genome sequences in two other forms: *coding region* and *protein*.
- Using contrasting or emerging pattern mining (Ventura & Luna, 2018) on the developed corpus to discover contrasting frequent patterns of nucleotides and using these patterns for the analysis and classification task.

CRediT author statement

M. Saqib Nawaz: Data Curation, Methodology, Validation, Visualization, Writing - Original Draft, Writing - Review & Editing. **Philippe Fournier-Viger:** Supervision, Formal analysis, Methodology, Validation, Writing - Review & Editing. **Shoaib Nawaz:** Investigation, Visualization, Writing - Review & Editing. **Haowei Zhu:** Data curation, Conceptualization, Formal analysis. **Unil Yun:** Investigation, Visualization, Writing - Review & Editing.

Conflict of Interest: Authors declare no conflict on interest.
Funding: Authors did not receive funding for this work.

References

Aggrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of Very Large Databases (VLDB)* (p. 487-499).

Ahamad, M. M., Aktar, S., Rashed-Al-Mahfuz, M., Uddin, S., Lio, P., Xu, H., ... Moni, M. A. (2020). A machine learning model to identify early stage symptoms of SARS-CoV-2 infected patients. *Expert Systems with Applications*, *160*, 113661. https://doi.org/10.1016/j.eswa.2020.113661

Ahmed, I., & Jeon, G. (2022). Enabling artificial intelligence for genome sequence analysis of COVID-19 and alike viruses. *Interdisciplinary Sciences: Computational Life Sciences*, *14*, 504-519. https://doi.org/10.1007/s12539-021-00465-0

Ali, S., Sahoo, B., Ullah, N., Zelikovskiy, A., Patterson, M., & Khan, I. (2021). A k-mer based approach for SARS-COV-2 variant identification. In *International Symposium on Bioinformatics Research and Applications (ISBRA)* (p. 153-164). https://doi.org/10.1007/978-3-030-91415-8_14

Alshayeji, M. H., Sindhu, S. C., & Abed, S. (2023). Viral genome prediction from raw human DNA sequence samples by combining natural language processing and machine learning techniques. *Expert Systems with Applications*, *218*, 119641. https://doi.org/10.1016/j.eswa.2023.119641

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*(3), 403-410. https://doi.org/10.1016/S0022-2836(05)80360-2

Arslan, H. (2021a). COVID-19 prediction based on genome similarity of human sars-cov-2 and bat sars-cov-like coronavirus. *Computers & Industrial Engineering*, *161*, 107666. https://doi.org/10.1016/j.cie.2021.107666

Arslan, H. (2021b). Machine learning methods for COVID-19 prediction using human genomic data. *Proceedings*, *74*(1), 20. https://doi.org/10.3390/proceedings2021074020

Arslan, H., & Arslan, H. (2021). A new COVID-19 detection method from human genome sequences using cpg island features and knn classifier. *Engineering Science and Technology, an International Journal*, *24*(4), 839-847. https://doi.org/10.1016/j.jestch.2020.12.026

Cellier, P., Charnois, T., Plantevit, M., Rigotti, C., Cremilleux, B., Gandrillon, O., ... Manguin, J.-L. (2013). Sequential pattern mining for discovering gene interactions and their contextual information from biomedical texts. *Journal of Biomedical Semantics*, *6*, 27. https://doi.org/10.1186/s13326-015-0023-3

Dlamini, G. S., Muller, S. J., Meraba, R. L., Young, R. A., Mashiyane, J., Chiwewe, T., & Mapiye, D. S. (2021). Classification of COVID-19 and other pathogenic sequences: A dinucleotide frequency and machine learning approach. *IEEE Access*, *8*, 195263-195273. https://doi.org/10.1109/ACCESS.2020.3031387

El-Dosuky, M. A., Soliman, M., & Hassanien, A. E. (2021). COVID-19 vs Influenza viruses: A cockroach optimized deep neural network classification approach. *International Journal of Imaging Systems and Technology*, *31*, 471-482. https://doi.org/10.1002/ima.22562

Exarchos, T. P., Papaloukas, C., Lampros, C., & Fotiadis, D. I. (2008). Mining sequential patterns for protein fold recognition. *Journal of Biomedical Informatics*, *41*(1), 165-179. https://doi.org/0/10.1016/j.jbi.2007.05.004

Fournier-Viger, P., Gomariz, A., Campos, M., & Thomas, R. (2014). Fast vertical mining of sequential patterns using co-occurrence information. In *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)* (p. 40-52). https://doi.org/10.1007/978-3-319-06608-0_4

Fournier-Viger, P., Gomariz, A., Gueniche, T., Mwamikazi, E., & Thomas, R. (2013). TKS: Efficient mining of top-k sequential patterns. In *Proceed-*

ings of Advanced Data Mining and Applications (ADMA) (p. 109-120). https://doi.org/10.1007/978-3-642-53914-5_10

Fournier-Viger, P., Lin, J. C.-W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., & Lam, H. T. (2016). The SPMF open-source data mining library version 2. In *Proceedings of ECML/PKDD* (p. 36-40). https://doi.org/10.1007/978-3-319-46131-1_8

Fournier-Viger, P., Lin, J. C.-W., Kiran, R. U., Koh, Y. S., & Thomas, R. (2017). A survey of sequential pattern mining. *Data Science and Pattern Recognition*, *1*, 54-77.

Frank, E., Hall, M. A., & Witten, I. H. (2016). *The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques*, fourth edition. Morgan Kaufmann.

Gunasekaran, H., Ramalakshmi, Arokiaraj, R. M., Kanmani, D., Venkatesan, C., & Dhas, C. S. G. (2021). Analysis of DNA sequence classification using CNN and hybrid models. *Computational and Mathematical Methods in Medicine*, *1835056*. https://doi.org/10.1155/2021/1835056

Hsu, C.-M., Chen, C.-Y., Hsu, C.-C., & Liu, B.-J. (2006). Efficient discovery of structural motifs from protein sequences with combination of flexible intra- and inter-block gap constraints. In *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)* (p. 530-539). https://doi.org/10.1007/11731139_62

Iqbal, M., Setiawan, M. N., Irawan, M. I., Khalif, K. M. N. K., Muhammad, N., & Aziz, M. K. B. M. (2022). Cardiovascular disease detection from high utility rare rule mining. *Artificial Intelligence in Medicine*, *131*, 102347. https://doi.org//10.1016/j.artmed.2022.102347

Jing, R., Li, Y., Xue, L., Liu, F., Li, M., & Luo, J. (2020). autoBioSeqpy: A deep learning tool for the classification of biological sequences. *Journal of Chemical Information and Modeling*, *60*(8), 3755-3764. https://doi.org/10.1021/acs.jcim.0c00409

Johnson, A. D. (2010). An extended iupac nomenclature code for polymorphic nucleic acids. *Bioinformatics*, *26*(10), 1386-1389. https://doi.org/10.1093/bioinformatics/btq098

Kalia, K., Saberwal, G., & Sharma, G. (2021). The lag in SARS-CoV-2 genome submissions to GISAID. *Nature Biotechnology*, *39*, 1058-1060. https://doi.org/10.1038/s41587-021-01040-0

Karim, M. R., Rashid, M. M., Jeong, B.-S., & Choi, H.-J. (2012). An efficient approach to mining maximal contiguous frequent patterns from large DNA sequence databases. *Genomics Informatics*, *10*, 51-57. https://doi.org/10.5808/GI.2012.10.1.51

Kuzmin, K., Adeniyi, A. E., Jr., A. K. D., Lim, D., Nguyen, H., Molina, N. R., ... Harrison, R. W. (2020). Machine learning methods accurately predict host specificity of coronaviruses based on spike sequences alone. *Biochemical and Biophysical Research Communications*, *533*(3), 553-558. https://doi.org/10.1016/j.bbrc.2020.09.010

Lopez-Rincon, A., Tonda, A., Mendoza-Maldonado, L., Mulders, D. G. J. C., Molenkamp, R., Perez-Romero, C. A., ... Kraneveld, A. D. (2021). Classifi-

cation and specific primer design for accurate detection of SARS-CoV-2 using deep learning. *Scientific Reports*, *11*, 947. https://doi.org/10.1038/s41598-020-80363-5

Mateos, P. A., Balboa, R. F., Easteal, S., Eyras, E., & Patel, H. R. (2021). PACIFIC: A lightweight deep-learning classifier of SARS-CoV-2 and co-infecting RNA viruses. *Scientific Reports*, *11*, 3209. https://doi.org/10.1038/s41598-021-82043-4

Members, C.-N., & Partners. (2023). Database resources of the national genomics data center, China national center for bioinformation in 2023. *Nucleic Acids Research*, *51*(D1), D18-D28. https://doi.org/10.1093/nar/gkac1073

Naeem, S. M., Mabrouk, M. S., Marzouk, S. Y., & Eldosoky, M. A. (2021). A diagnostic genomic signal processing (GSP)-based system for automatic feature analysis and detection of COVID-19. *Briefings in Bioinformatics*, *2*(2), 1197-1205. https://doi.org/10.1093/bib/bbaa170

Nawaz, M. S., Fournier-Viger, P., Aslam, M., Li, W., He, Y., & Niu, X. (2023). Using alignment-free and pattern mining methods for SARS-CoV-2 genome analysis. *Applied Intelligence*, *53*, 21920-21943. https://doi.org/10.1007/s10489-023-04618-0

Nawaz, M. S., Fournier-Viger, P., & He, Y. (2022). S-PDB: Analysis and classification of SARS-CoV-2 spike protein structures. In *Proceedings of international conference on bioinformatics and biomedicine (BIBM)* (p. 2259-2265). https://doi.org/10.1109/BIBM55620.2022.9995562

Nawaz, M. S., Fournier-Viger, P., He, Y., & Zhang, Q. (2023). PSAC-PDB: Analysis and classification of protein structures. *Computers in Biology and Medicine*, *158*, 106814. https://doi.org/10.1016/j.compbiomed.2023.106814

Nawaz, M. S., Fournier-Viger, P., Nawaz, M. Z., Chen, G., & Wu, Y. (2022). MalSPM: Metamorphic malware behavior analysis and classification using sequential pattern mining. *Computers & Security*, *118*, 102741. https://doi.org/10.1016/j.cose.2022.102741

Nawaz, M. S., Fournier-Viger, P., Shojaee, A., & Fujita, H. (2021). Using artificial intelligence techniques for covid-19 genome analysis. *Applied Intelligence*, *53*, 3086-3103. https://doi.org/10.1007/s10489-021-02193-w

Pearson, W. R. (1994). Using the FASTA program to search protein and DNA sequence databases. *Methods in Molecular Biology*, *24*, 307-331. https://doi.org/10.1385/0-89603-246-9:307

Poor, N. R., & Yaghoobi, M. (2019). A new approach in DNA sequence compression: Fast DNA sequence compression using parallel chaos game representation. *Expert Systems with Applications*, *116*, 487-493. https://doi.org/10.1016/j.eswa.2018.09.012

Qiang, X.-L., Xu, P., Fang, G., Liu, W.-B., & Kou, Z. (2020). Using the spike protein feature to predict infection risk and monitor the evolutionary dynamic of coronavirus. *Infectious Diseases of Poverty*, *9*, 33. https://doi.org/10.1186/s40249-020-00649-8

Randhawa, G. S., Soltysiak, M. P. M., Roz, H. E., de Souza, C. P. E., Hill, K. A., & Kari, L. (2020). Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study. *PLoS One*, *15*(4),

e0232391. https://doi.org/10.1371/journal.pone.0232391

Roberts, M., Hunt, B. R., Yorke, J. A., Bolanos, R. A., & Delcher, A. L. (2004). A preprocessor for shotgun assembly of large genomes. *Journal of Computational Biology*, *11*, 734–752. https://doi.org/10.1089/cmb.2004.11.734

Sallaberry, A., Pecheur, N., Bringay, S., Roche, M., & Teisseire, M. (2011). Sequential patterns mining and gene sequence visualization to discover novelty from microarray data. *Journal of Biomedical Informatics*, *44*(5), 760-774. https://doi.org/10.1016/j.jbi.2011.04.002

Sayers, E. W., Cavanaugh, M., Clark, K., Ostell, J., Pruitt, K. D., & Karsch-Mizrachi, I. (2020). Genbank. *Nucleic Acids Research*, *48*, D84-D86. https://doi.org/10.1093/nar/gkz956

Singh, O. P., Vallejo, M., El-Badawy, I. M., Aysha, A., Madhanagopal, J., & Faudzi, A. A. M. (2021). Classification of SARS-CoV-2 and non-SARS-CoV-2 using machine learning algorithms. *Computers in Biology and Medicine*, *136*, 104650. https://doi.org/10.1016/j.compbiomed.2021.104650

Ventura, S., & Luna, J. (2018). *Supervised Descriptive Pattern Mining*. Springer. https://doi.org/10.1007/978-3-319-98140-6

Vinga, S. (2013). Information theory applications for biological sequence analysis. *Briefings in Bioinformatics*, *15*(3), 376-389. https://doi.org/10.1093/bib/bbt068

Wang, M., Shang, X.-Q., & Li, Z.-H. (2008). Sequential pattern mining for protein function prediction. In *Proceedings of Advanced Data Mining and Applications (ADMA)* (p. 652-658). https://doi.org/10.1007/978-3-540-88192-6_68

Wu, F., Zhao, S., Yu, B., Chen, Y.-M., Wang, W., Song, Z.-G., ... Zhang, Y.-Z. (2020). A new coronavirus associated with human respiratory disease in China. *Nature*, *579*(7798), 265-269. https://doi.org/10.1038/s41586-020-2008-3

Wu, X., Zhu, X., He, Y., & Arslan, A. N. (2013). PMBC: Pattern mining from biological sequences with wildcard constraints. *Computers in Biology and Medicine*, *43*(5), 481-492. https://doi.org/10.1016/j.compbiomed.2013.02.006

Ye, S. H., Siddle, K. J., Park, D. J., & Sabeti, P. C. (2019). Benchmarking metagenomics tools for taxonomic classification. *Cell*, *178*, 779-794. https://doi.org/10.1016/j.cell.2019.07.010

Zielezinski, A., Vinga, S., Almeida, J., & Karlowski, W. M. (2017). Alignment-free sequence comparison: Benefits, applications, and tools. *Genome Biology*, *18*, 1-17. https://doi.org/10.1186/s13059-017-1319-7

Zihayat, M., Davoudi, H., & An, A. (2017). Mining significant high utility gene regulation sequential patterns. *BMC Systems Biology*, *11*, 109. https://doi.org/10.1186/s12918-017-0475-4

Appendix

ACC is defined as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP = true positive: total count of frequent sequential patterns of nucleotides that are correctly classified to a particular virus class.

TN = the true negative: total count of frequent sequential patterns of nucleotides that is correctly identified as not belonging to a particular virus class.

FP = false positive: total count of frequent sequential patterns of nucleotides that are incorrectly classified to a particular virus class, and

FN = false negative: total count of frequent sequential patterns of nucleotides that is incorrectly classified as not belonging to a given virus class.

The following formulas are used for other five measures:

$$FPR = \frac{FP}{FP + TN}$$

$$Recall(R) = \frac{TP}{TP + FN}$$

$$Precision(P) = \frac{TP}{TP + FP}$$

$$F - measure = 2 \times \frac{P \times R}{P + R}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$