Applied Intelligence (2023) 53:21920–21943
https://doi.org/10.1007/s10489-023-04618-0

Using alignment-free and pattern mining methods for SARS-CoV-2 genome analysis

M. Saqib Nawaz[1] · Philippe Fournier-Viger[1] · Memoona Aslam[2] · Wenjin Li[2] · Yulin He[3] · Xinzheng Niu[4]

Accepted: 7 April 2023 / Published online: 16 June 2023

Abstract

Examining the genome sequences of the SARS-CoV-2 virus, that causes the respiratory disease known as coronavirus disease 2019 (COVID-19), play important role in the proper understanding of this virus, its main characteristics and functionalities. This paper investigates the use of alignment-free (AF) sequence analysis and sequential pattern mining (SPM) to analyze SARS-CoV-2 genome sequences and learn interesting information about them respectively. AF methods are used to find (dis)similarity in the genome sequences of SARS-CoV-2 by using various distance measures, to compare the performance of these measures and to construct the phylogenetic trees. SPM algorithms are used to discover frequent amino acid patterns and their relationship with each other and to predict the amino acid(s) by using various sequence-based prediction models. In last, an algorithm is proposed to analyze mutation in genome sequences. The algorithm finds the locations for changed amino acid(s) in the genome sequences and computes the mutation rate. From obtained results, it is found that that both AF and SPM methods can be used to discover interesting information/patterns in SARS-CoV-2 genome sequences for examining the variations and evolution among strains.

Keywords COVID-19 · SARS-CoV-2 · Genome sequence · Amino acids · Alignment-free · Sequential pattern mining · Mutation

1 Introduction

The COVID-19(novel coronavirus 2019) [1] disease is caused by the virus officially known as Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) [2]. For a disease, particularly for a pandemic, finding its genome characteristics (or features) and performing experiments allow biomedical experts to come with a hypothesis on the effect of these features on the disease's manifestations. However, this process is not only slow but also resource intensive and requires following certain safety protocols. Computational and computer-assisted studies, on the other hand, are fast and can be performed easily. They provide important information that is sometimes challenging to obtain from wet-lab experiments. Thus, we believe that sequence alignment [3] and frequent pattern mining [4] can be used to find actionable insights that may provide a better global response.

The sequence alignment field in bioinformatics deals with comparing and finding (dis)similarities between biological sequences. Now, various alignment-free (AF) approaches for sequence comparison and analysis are available [5–8]. Over the years, these approaches have emerged as a natural framework to understand the patterns and important characteristics and properties of biological sequences. AF approaches are based on the principle of converting the symbolic biological sequences into vectors spaces. This conversion enables one to efficiently use various filtering, normalization, (dis)similarity calculation and clustering techniques on biological sequences. AF approaches are used in the phylogenetic study, regulatory elements, sequence assembly and protein classification. However, their applicability and potential for the analysis of SARS-CoV-2 genome sequence needs further investigation and exploration.

Similarly, the pattern mining field is also used in the analysis of complex and large genetic and genomic data. In pattern mining, sequential pattern mining (SPM) [9] is a special case of structured data mining that has been used not only in genomics [10–12] but also in other areas such as market

✉ Philippe Fournier-Viger
philfv@szu.edu.cn

Extended author information available on the last page of the article

basket analysis [13], text analysis [14, 15], proof sequence learning [16, 17], energy reduction in smarthomes [18], malware detection [19] and webpage click-stream analysis [20]. For genome data, SPM can provide new and useful insights related to virus behavior, severity (virulence) and other disease manifestations. Moreover, using SPM in genomes to discover important hidden information can help in speeding up the biological research process and can be of great importance to the biological world.

AF methods and SPM were used in some early studies [12, 21] to analyze and compare SARS-CoV-2 genome sequences, and to discover hidden interesting patterns of nucleotides and their prediction(s) in those sequences. The SARS-CoV-2 genome sequences in *nucleotide form* containing four nucleotides (Adenine-A, Guanine- G, Cytosine-C and Thymine-T) were considered in both studies. In this work, we explore the use of AF methods and SPM on SARS-CoV-2 genome sequences in *protein form*. Proteins are final products of gene expression which consist of long chains of amino acids (AAs). More specifically, this paper extends the authors' previous work [12, 21]. Following are the main contributions of this paper:

1. **Investigating various AF methods to analyze SARS-CoV-2 genome sequences in protein form:** More specifically, two tools that implement various AF methods are used to (1) find (dis)similarity by using various distance measures, (2) compare the performance of these measures, and (3) examine and explore various AF methods in the phylogenetic tree as well as the consensus tree construction.
2. **Using SPM to analyze SARS-CoV-2 genome sequences:** SPM algorithms are used to (1) discover frequent AA patterns and their relationship with each other, and (2) predict the AA(s) by using various state-of-the-art sequence prediction models.
3. **Mutation analysis in SARS-CoV-2 genome sequences:** An algorithm is proposed for point-wise mutation analysis of AAs and to compute the mutation rate.

The remainder of this paper is organized as follows. Background on SARS-CoV-2 is provided in Section 2, followed by a review of computational and computer-assisted studies about COVID-19 in Section 3. The proposed approaches where AF methods and SPM are used to analyze SARS-CoV-2 genome sequences are presented in Section 4. Section 5 discusses the evaluation of the proposed approaches and obtained results. The proposed mutation analysis technique is described in Section 6, followed by a conclusion in Section 7.

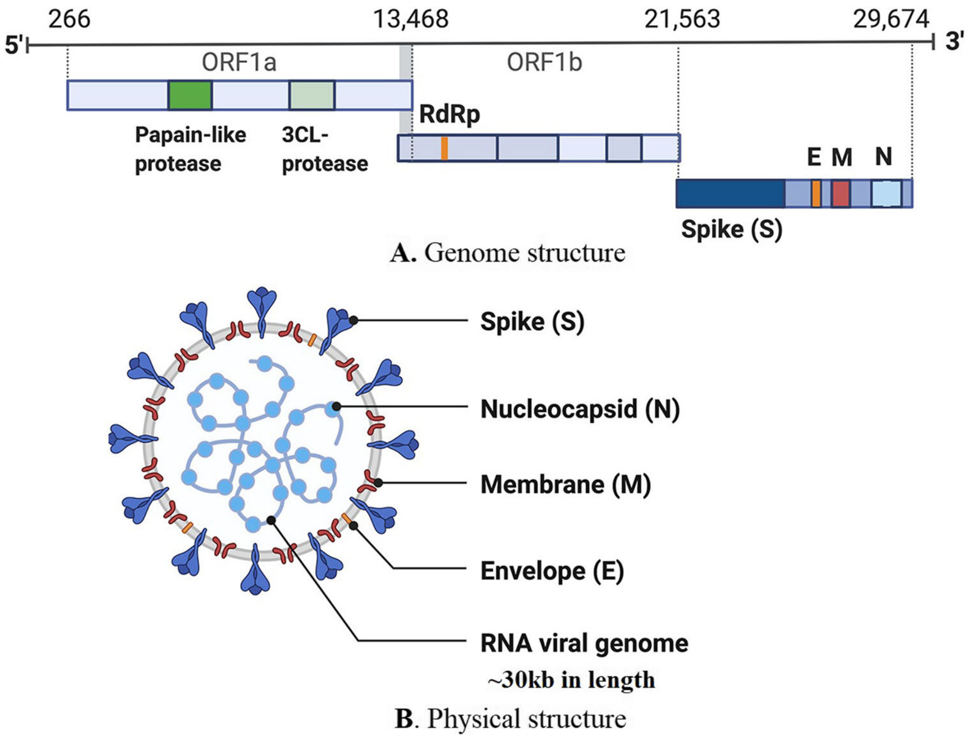
2 SARS-CoV-2 background

The genome sequences of a virus can be made from DNA or RNA. SARS-CoV-2 is from the family of beta-coronavirus with non-segmented positive-sense, single-stranded ribonucleic acid (RNA) having spherical to pleomorphic shape and a length between 80 to 160 nm. This virus is made from some non-structural proteins and four structural proteins, that are S: Spike, E: Envelope, M: Membrane and N: Nucleocapsid (Fig. 1). The outer layer of this virus is made from three structural proteins, S, M, and E. E is also involved in the maturation and production of this virus. Whereas, S and M also play important role in the virus attachment process during replication. The nucleocapsid inside is formed by the N protein.

This virus can enter in the human body by interacting with the host angiotensin-converting enzyme 2 (ACE2). Note that ACE2 receptor can be found in various organs inside the human body such as kidneys, heart, lungs and the gastrointestinal tract. S1 and S2, the two sub-units of the S protein, are mainly responsible for the receptor binding in the host cells [23]. After the S protein binds with the receptor, the envelope fuses with the cell membrane. Thus, this virus genome is released into the target cell.

SARS-CoV-2 releases its genomic material as mRNA. The genome contains about six to twelve open reading frames (ORFs) and its size is in the range of 29.8 kb to 30 kb. At the 5'untranslated region (UTR), approximately two-thirds of the genome consists of the ORF1a and ORF1b polyproteins. At the 3'UTR, one third of the genome comprises the four structural proteins. Some accessory proteins such as ORF3a, ORF6, ORF7ab, ORF8 and ORF10 are also present in SARS-CoV-2 [24].

SARS-CoV-2 genome sequence contains four main nucleotide bases (Adenine-A, Cytosine-C, Guanine-G, and Thymine-T) that appear in a specific order. A sequence of three nucleotide bases is called a codon. As there are 4 nulceotides, so there are $4^3 = 64$ different codons in total. From these 64 codons, 61 represent different amino acids (AAs) and the remaining three codons play the role of stop signals. As there are 61 codons but only 20 different AAs, more than one codon is used to encode different AAs. For mapping codons to AAs, the genetic code is used, where one codon that contains three nucleotides encodes one AA [25]. In genome sequences, the term *k-mers* represents unique subsequences of length k. For example, there are four *k-mers* for $k = 1$, which are A, C, G and T. The sequence $GCCTA$ contains four 2-$mers$ (GC, CC, CT, and TA) and three 3-*mers* (GCC, CCT and CTA). A sample of SARS-CoV-2 genome sequences is shown in Fig. 2. Each sequence (row)



A. Genome structure

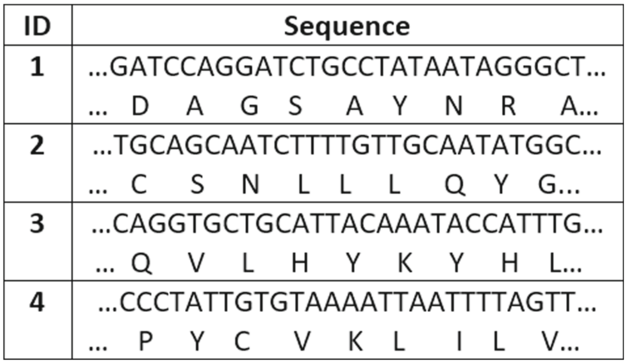
B. Physical structure

Fig. 1 SARS-CoV-2 structure[22]

is represented in *nucleotide form* (top) and *protein form* (bottom). In the first genome sequence with ID 1, the first three nucleotides are G, A and T, that is the codon GAT, which encodes the Aspartate (D) amino acid.

3 Related work

The reviews [26–31] provide a comprehensive overview on the use of artificial intelligence (AI), machine learning (ML) and deep learning in COVID-19 studies. These methods have been utilized in the past mostly for medical imaging (such as computed tomography (CT) and X-Ray) segmentation and diagnosis of COVID 19 [32]. However, as Driggs et al. argued in an editorial [33], almost all the machine learning based studies for COVID-19 diagnosis or prognosis did not follow any standard approach for development or evaluation. This makes it very hard, even for the experts, to select the model that may provide the most clinical benefits. Roberts et al. [34] reviewed 62 published articles for COVID-19 diagnosis or prognosis from CXR or CT images. It was found that models proposed in these articles have no potential clinical use because of underlying biases and/or methodological flaws. Similarly, 169 studies that described prediction models for COVID-19 were reviewed by Wynants et al. [35], which concluded that prediction models are badly reported and highly biased. They recommended and suggested that well documented data from COVID-19 studies should be used in prediction models and methodological guidance must be followed to develop reliable prediction models.



ID	Sequence
1	...GATCCAGGATCTGCCTATAATAGGGCT... ... D A G S A Y N R A...
2	...TGCAGCAATCTTTTGTTGCAATATGGC... ... C S N L L L Q Y G...
3	...CAGGTGCTGCATTACAAATACCATTTG... ... Q V L H Y K Y H L...
4	...CCCTATTGTGTAAAATTAATTTTAGTT... ... P Y C V K L I L V...

Fig. 2 SARS-CoV-2 genome sequences represented as nucleotides and AAs

Noor et al. [36] performed a thematic analysis of COVID-19 related tweets using the VOSviewer software to evaluate public reactions towards the pandemic. SPM algorithms were also used in that study to discover frequent words/patterns in tweets and their relationships with each other. Heng et al. [37] compared three SARS-CoV-2 genome sequences by using two pairwise sequence alignment (PSA) algorithms known as the Needleman-Wunsch and Smith-Waterman algorithms for mutation analysis. Pathan et al. [38] computed the mutation rate in the genome sequences of COVID-19. They calculated the codon mutation and missense nucleotide mutation rates. Moreover, a deep learning method was used for predicting the future mutation rate of SARS-CoV-2. The study

[39] analyzed 329,942 SARS-CoV-2 genome sequence that were downloaded from the GISAID [40] website, excluding the ORF1ab gene of sequences. They found 155 single nucleotide polymorphsim (SNP) in more than 0.3% of the sequences. Moreover, clustering results showed the existence of B.1.1.7 (Alpha) variant subtype and the two most conserved genes were E and ORF6.

For SARS-CoV-2 genome sequence classification and detection, two CpG-based features have been used [41, 42]. Four classifiers were compared [41] and it was found that the random forest classifier was more accurate than SVM (Support Vector Machines, NB (Naive Bayes) and kNN (k-Nearest Neighbors).

The performance of kNN was also investigated by using 19 distance measures under five categories [42]. Besides, three similarity features have been integrated with the two CpG-based features to improve SARS-COV-2 genome sequence classification and prediction [43]. We believe that labeling genome sequences for these studies [41–43] was done manually, which consumes a lot of time. Moreover, authors claim that their proposed approach for COVID-19 genome sequence detection is fast. However, no experiments are performed in this regard. Representative genomic sequences of SARS-CoV-2 were discovered by Lopez-Rincon et al. [44] by coupling a deep learning method with explainable AI techniques. In another study [45], the discrete Cosine and Fourier transforms and various moment invariants were utilized for feature extraction from 76 SARS-CoV-2 genome sequences. Moreover, kNN and a cascade-forward back propagation neural network were used for classification, where kNN performed better. In both [44, 45], the dataset contain few sequences and their feature extraction method are expensive. Randhawa et al. [46] designed a classification method using an intrinsic genomic signature found in SARS-CoV-2 with a machine learning-based AF method for classification of SARS-CoV-2 genome sequences. Ahmed and Jeon [47] compared genome sequences of four viruses, SARS-CoV-1, SARS-CoV-2, MERS and Ebola by using various analysis techniques. ML algorithms were also used for the classification of genome sequences.

4 Analyzing SARS-CoV-2 genome sequences with AF and SPM methods

This section provides the detail of the proposed approach for the first two contributions, where AF methods and SPM are used for the analysis of SARS-CoV-2 genome sequences that contain AAs. The approach consists of two main parts:

1. Various AF methods are first used to discover frequent AA patterns and identify (dis)similarities between SARS-CoV-2 genome sequences. Various distance measures performance is also compared. Moreover, various AF methods are used to construct the phylogenetic tree of SARS-CoV-2 genome sequences and the consensus tree.
2. The AAs present in the genome sequences of SARS-CoV-2 are first transformed into integers. SPM algorithms are then used to not only find the AAs that occur frequently, but also the sequential relationships between AAs. Moreover, various sequence-based prediction models are used and compared for the prediction of the next AA in a sequence(s).

More details on the two parts are provided next.

4.1 AF methods

An overview on AF methods are provided in this section that are used to analyze the genome sequences of SARS-CoV-2. AF methods can be divided into two main categories, that are:

1. Word-based methods: Find the total occurrence of word patterns (*k-mers* or *k-tuples*) in genomic sequences and compare sequences using similarity/dissimilarity measures based on *k-mer* frequencies.
2. Information theory-based methods: Find and compute the information shared among genomic sequences.

Besides the aforementioned two categories, some other AF methods are based on the common substrings length, iterated maps, Fourier-transformation, sequence representation based on chaos theory, micro-alignments and nucleotides positions moments, etc. More details about AF methods and their categories can be found in other studies [5–8, 48–50]. From a mathematical point of view, all AF methods are well founded in the areas of linear algebra, probability, statistics and information theory. Pairwise measures are generally used to calculate similarity/dissimilarity or distance among sequences.

4.1.1 AF methods based on word/k-mer

These AF methods compute the similarity/dissimilarity in genome sequences on the basis of occurrences of all *k-mers*. These methods are based on the concept that similar words/*k-mers* are present in similar sequences and using mathematical operations on the occurrences of *k-mers* give a good measure to compute similarity/dissimilarity between sequences. Over the years, various similarity/dissimilarity measures have been proposed and developed. These measures can be divided into two main groups: (1) measures that do not need background word frequencies and (2) measures that require background word frequencies. For the first type of measures, the observed word count/frequency or word presence/absence are used to find the similarity/dissimilarity measures. Thus, these measures are based on *k-mer* counts/frequencies and measures based on presence/absence of *k-mers*. The three main steps for word count/frequency-based AF methods (Fig. 3(a)) are [5]:

1. Divide sequences into *k-mers*: The sequences are first divided into unique words of a given length (*k-mer*). A simple example is provided for explanation. Let $x = ATGTGTG$ and $y = CATGTG$ are two sequences. For *3-mer* that contain 3 elements, x and y are cut into: W_3^x =$\{ATG, TGT, GTG, TGT, GTG\}$ and W_3^y = $\{CAT, ATG, TGT, GTG\}$. Note that three unique words (ATG, TGT, GTG) are present in W_3^x. Sets of words present in both W_3^x and W_3^y are combined using the union to generate the full word set W^3 =$\{ATG, CAT, GTG, TGT\}$. Note that the words present in W^3 belong to either W_3^x or W_3^y.
2. Transform sequences into vectors: The word set obtained after splicing are transformed into vectors. Take the sequence x as example. The vector for x contains the number of occurrences for each particular *k-mer* (from W^3) appearance in x. Thus for two sequences (x and y), the two created vectors are: c_3^x = (1, 0, 2, 2) and c_3^y =(1, 1, 1, 1).
3. Apply distance functions to find similarity/dissimilarity: The similarity/dissimilarity among sequences is calculated by applying a distance function on the vectors c_3^x and c_3^y. For example, the Euclidean (EU) distance can be used to find the similarity/dissimilarity [51]:

   $$Eu(x, y) = \sqrt{\sum_{w \in A^k} (f_w^{(x)} - f_w^{(y)})^2}$$

   where A^k denotes the set of all *k-mers* that are present in two sequences and $f_w^{(x)}$ and $f_w^{(y)}$ denotes the occurrence frequencies of a *k-mer* w in the two sequences x and y.

   A high similarity/dissimilarity value shows that the sequences are more similar/distant. In word-based AF methods, the conversion of sequences into vectors gives one the advantage of using various distance measures (such as Manhattan distance, Chebyshev distance, Euclidean distance, Canberra distance, to name a few) to compute the similarity/dissimilarity.

For measures that are based on the presence/absence of words, the *k-mers* are treated as binary data. For two sequences $G^{(1)}$ and $G^{(2)}$, the Hamming distance can be calculated as [51]:

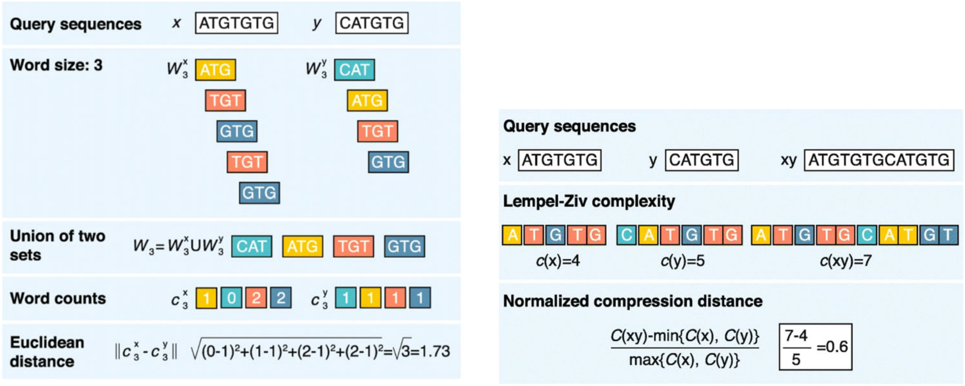
$$Hamming = (B + C)/N$$

where B denotes the *k-mers* present in $G^{(1)}$ that are absent from $G^{(2)}$, C denotes the opposite of B and N indicates the total number of *k-mers*.

In the measures that do not require background word frequencies (CVTree, d_2 and d_2^S), the discovered word counts/frequencies or word presence/absence are used directly to evaluate similarity/dissimilarity. Calculating these measures also requires obtaining knowledge about the approximate distribution of word counts/frequencies in the background sequences, for which Markov chains (MC) [52] are widely used.

4.1.2 AF methods based on information theory

For AF sequence analysis, the field of information theory has provided successful methods. They can find and compute the amount of information shared among two sequences. Genomic sequences are made from nucleotides and AAs. Both of them are basically strings of symbols. Thus, information theory metrics such as entropy and complexity can naturally interpret the digital organization of nucleotides and AAs.



(a) Word-based

(b) Information theory-based

Fig. 3 Word and information theory-based distances calculation [5]

The Kolmogorov complexity of genome sequences can be calculated as the length of their shortest descriptions. Naturally, sequences having longer descriptions generally show a higher complexity. Thus, the Kolmogorov complexity in this case cannot discover the shortest description for a given character string. To solve this problem, general compression algorithms can be used for complexity approximation. With such algorithms, the compressed sequence length offers a good estimate for the complexity. Thus, a complex string will be less compressible. Computing the distance between sequences by using complexity (also called compression) involves three main steps: (Fig. 3(b)).

1. The sequences under consideration ($x = ATGTGTG$ and $y = CATGTG$) are combined to generate a long sequence ($xy = ATGTGTGCATGTG$).
2. Complexity calculation. If x and y are exactly the same, then xy complexity will be very close either to x or y. If both sequences (x and y) are not same, then xy complexity will be close to their cumulative complexities. Various information-based distance functions can be found in the literature. For example, the Lempel-Ziv complexity [53] finds the different subsequences that are observed when a sequence is read from start to end (Fig. 3(b)).
3. Normalized compression distance (NCD) [54], which is a compressed distance measure, can be used to find the similarity/difference between sequences:

   $$NCD(x, y) = \frac{C(xy) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}}$$

   where the compressor such as bzip2 or gzip is represented with C.

A popular entropy-based measure is the Kullback-Leibler divergence, which is another information measurement that can be used for sequence comparison. The comparison process involves (1) finding the frequencies/counts of symbols or words in a sequence, and (2) summing their entropies in the compared sequences. Similarly, the Base-base correlation (BBC) measure offers a novel sequence feature for studying the genome information structure. This measure has been used for the differentiation of various functional regions of genomes. A genome sequence is converted in this measure into a unique 16-dimensional numeric vector by using the following equation [49]:

$$T_{ij}(K) = \sum_{n=1}^{K} P_{ij}(n).log_2\left(\frac{P_{ij}(n)}{P_i P_j}\right)$$

where P_i and P_j are the probabilities of nucleotides i and j, $P_{ij}(n)$ is the probability of nucleotides (i and j) at distance n in the genome and K denotes the maximum distance between i and j. Interested readers can read more about compression algorithms for AF methods in [55].

In this work, we used two tools Alfree [5] and CAFE [51] that implement various AF methods. Methods that are implemented in Alfree compute the distances among sequences by finding various patterns and their properties in sequences that are unaligned. Alfree provides implementation for 38 AF methods. These methods can be used to calculate the distances among nucleotides, amino acids or protein sequences, and for tree construction. The main feature of Alfree is that one can create consensus phylogenetic trees. These trees give a good estimation for the support level (agreement) among trees obtained by various individual methods. Through this tree, one can examine the reliability of phylogenetic relationships between methods. The word and information theory-based AF methods implemented in Alfree are listed in Table 1. Alfree also implements methods that are based on graphical representations. Such methods are not discussed here as they accept only DNA as input.

The CAFE (aCcelerated Alignment-FrEe sequence analysis) [51] tool offers the platform for AF sequence comparison to study the relationships between genomes and

Table 1 AF methods in Alfree

Method	Distance
Word-based methods	
Euclidean distance	d^{S}, d^{E}, d^{Eseq1}, d^{Eseq2}
Minkowski distance	d^{Minkowski}
Absolute-based metrics	d$^{abs\_mean}$, d$^{abs\_mult}$, d^{Manhattan}, d^{Canberra}
Absolute-based metrics	d$^{abs\_mult1}$, d$^{abs\_mult2}$, d$^{Bray-Curtis}$, d^{Chebyshev}
Angle metrics	d^{EVOL1}, d^{EVOL2}
Composition distance	d^{CV}
Feature Frequency Profiles	d^{FFP}
Normalized Google Distance	d^{Google}
Linear Correlation Coefficient	d^{LCC}
Return Time Distribution	d^{RTD}
Boolean vectors	d^{Jaccard}, d^{Hamming}, d$^{Sorensen-Dice}$
Frequency Chaos Game Repr.	d^{FCGR}
Information Theory-based methods	
Lempel-Ziv complexity	d^{LZ}, d$^{LZ}_{*}$, d$^{LZ}_{1}$, d$^{LZ}_{*1}$, d$^{LZ}_{**1}$
NCD	d^{NCD}
Base-Base Correlation	d^{BBC}

meta-genomes. CAFE has a user-friendly GUI (graphical user interface) and in total, 28 AF measures are implemented in CAFE. Out of 28, 10 distance measures are based on *k-mer* counts/frequencies, 15 measures are based on absence/presence of *k-mers*, and 3 measures are based on background adjusted *k-mer* counts. In CAFE, one can see the results for distance measures in different visualizations that include dendrogram, heatmap, principal coordinate analysis (PCoA) and network analysis display. Table 2 lists the word-based AF methods that are implemented in CAFE.

4.2 SPM-based approach to analyze SARS-CoV-2 genome sequences

Genome sequences are built from nucleotides that make up AAs, which are generally strings of characters. Thus, SPM techniques are selected for their analysis.

The overall proposed learning approach with SPM and sequence prediction models is shown in Fig. 4. The approach has two parts:

1. **Development of Corpus**: Genome sequences of SARS-CoV-2 are converted into a corpus of discrete sequences, where AAs in the whole sequence are represented with distinct integers.
2. **Applying SPM and Sequence Prediction for Learning**: SPM algorithms are used on the corpus to not only discover AAs that occur frequently, but also the sequential relationships between them. Moreover, prediction techniques are used to predict the next amino acid(s) in a genome sequence.

Table 2 AF methods in CAFE

Word-based methods	
Chebyshev	Euclidean
Manhattan	Canberra
d_2 or cosine	Pearson
Feature Frequency Profiles	Jensen-Shannon divergence
Co-phylog	
Background adjusted word-based methods	
CVTree	d_2^*
d_2^S	
Presence/absence of words	
Gower	Kulczynski
Dice	Anderberg
Russel	Tanimoto
Jaccard	Antidice
Yule	Ochiai
Phi	Hamman
Sneath	Hamming
Matching	

More details on the two parts are provided next.

4.2.1 Corpus development

The online database GenBank [56] was used to obtain the genome sequences of SARS-CoV-2 strains. National Center for Biotechnology Information (NCBI) supports the GenBank and at the time of this paper submission, the NCBI database for SARS-CoV-2[1] contains 6,728,463 nucleotide records for SARS-CoV-2.

To efficiently use SPM and sequence prediction models, genome sequence data is first transformed into a suitable electronic format where the "*AAs to integers*" abstraction is used. Thus, each AA is converted into a distinct positive integer. This general abstraction allows one to use SPM algorithms and sequence prediction models to discover interesting and important patterns in the corpus and perform accurate prediction respectively.

The SARS-CoV-2 genome sequences corpus acquired from the GenBank [56] in *protein form* lists genome sequences in a FASTA format that contain the genes names, followed by AAs sequence. In the pre-processing step, the genes field in the genome sequences are removed. Thus, the complete genome sequence is a sequence of AAs. Combining all these AAs generates a corpus that has discrete sequences.

Let $AAs = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$ be the set of all distinct amino acids in sequences. The alphabets in the set represent different amino acids such as A and C represent the *Alanine* and *Cysteine* amino acids, respectively. $|AAs|$ represents the set cardinality of AAs. Thus, $|AAs| = 20$ as there are 20 distinct amino acids that make the proteins.

Now, a COVID-19 (SARS-CoV-2) genome sequence, represented as SGS, is an ordered list of amino acids, $SGS = \langle AAs_1, AAs_2, ..., AAs_n \rangle$, such that $AAs_i \subseteq AAs$ $(1 \leq i \leq n)$. Similarly, the *SARS-CoV-2 genome sequences corpus*, represented as $SGSC$, is a list of genome sequences $SGSC = \langle SGS_1, SGS_2, ..., SGS_p \rangle$. As an example, a $SGSC$ containing four lines (genome sequences) with four identifiers (IDs) is shown in Table 3.

As the genome sequences of SARS-CoV-2 in *protein form* contain sequences of AAs, each AA in the final step is replaced by a distinct positive integer. For example, the amino acid *Alanine* (A), Cysteine (C), Aspartate (D) and Glutamate (E) are replaced by 1, 2, 3 and 4 respectively. Moreover, for some SPM algorithms, -1 should be added between each AA and the row (line) must end with -2 [57].

[1] https://www.ncbi.nlm.nih.gov/sars-cov-2/

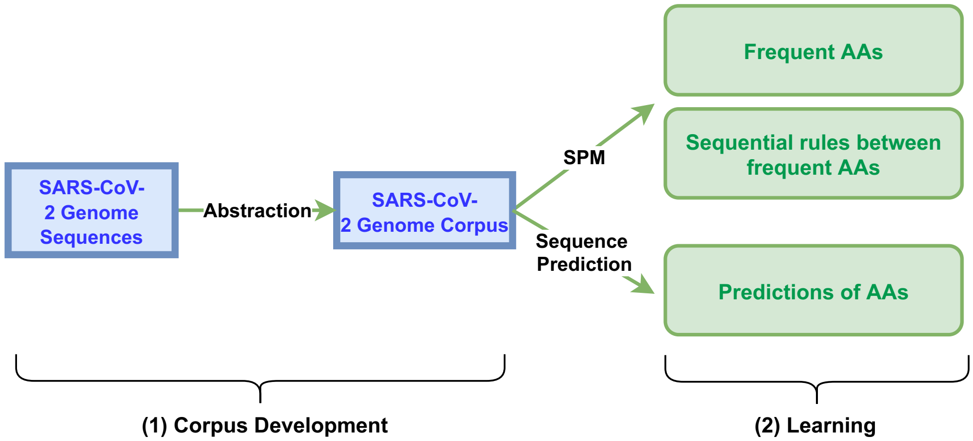


Fig. 4 The approach based on SPM and sequence prediction to analyze AAs in the genome sequences of SARS-CoV-2

4.2.2 Applying SPM and sequence prediction for learning

SPM techniques can find patterns (subsequences of AAs) that are present in genome sequences. A sequence, genome sequence in this work, $S_\alpha = \langle\, \alpha_1, \alpha_2, ..., \alpha_n \,\rangle$ is present (or contained) in another sequence $S_\beta = \langle \beta_1, \beta_1, ..., \beta_m \rangle$ iff there exist integers $1 \le i_1 < i_2 < ... < i_n \le m$, such that $\alpha_1 \subseteq \beta_{i1}, \alpha_2 \subseteq \beta_{i2}, ..., \alpha_n \subseteq \beta_{im}$ (denoted as $S_\alpha \sqsubseteq S_\beta$). The sequences S_α is a *subsequence* of S_β if S_α is contained in S_β. The *support* of S_α, denoted as $sup(S_\alpha)$, in a corpus $SGSC$ refers to the total number of sequences containing S_α. It is is defined as: $sup(S_\alpha) = |\{S | S_\alpha \sqsubseteq S \wedge S \in SGSC\}|$. For a $SGSC$, frequent SPM deals with identifying all the *frequent genome subsequences*. For a user-defined minimum support threshold $minsup$ ($minsup > 0$) and a $SGSC$, a genome subsequence S is frequent if $sup(S) \ge minsup$. Take Table 3 as an example, the subsequence $\langle CEF \rangle$ that contains three AAs has a support of 4 as the subsequence $\langle CEF \rangle$ is present in four lines.

The task of frequent SPM in a corpus that contains genome sequence of SARS-CoV-2 is not an easy task. The main reason is that genome sequences are generally similar to each other, with little difference, and they are very long. A genome sequence that contains n items (AAs in this work) can have up to $2^n - 1$ distinct subsequences. Thus, using the naive approach to calculate the support of all subsequences is infeasible. However, efficient algorithms have been proposed and developed in recent years that use various optimizations for finding the exact solution for a SPM problem without the need to explore the whole search space. SPM algorithms explore the pattern search space by first finding all frequent subsequences that contain 1 item (AA), called 1-sequences. Then, more items are appended to these subsequences recursively to find larger subsequences. Two operations (*s-extensions* and *i-extensions*) are used to do this. Both *s-extensions* and *i-extensions* are used to produce a $(k + 1)$-sequence from one or more k-sequences. Note that SPM can be used on more general cases where simultaneous items are allowed in a sequence. However, in this work, AAs are always totally ordered.

Table 3 A sample of a $SGSC$

ID	Sequence
1	⟨....WQTGDFVKANCEFCGTENLTKEGATTCGYL.....⟩
2	⟨....LEILQKEKVNINIVGDFKLNCEFNEIIILASF.....⟩
3	⟨....DGISQYSLRLIDAMMFTSDLATNNCEFXMAY.....⟩
4	⟨....TNCEFTLGGAPTKVTFGDDTVIEVQGYKSVN.....⟩

In SPM, it is necessary to define a total order relation $\prec$ on items so that sequential patterns can be identified rapidly and also to avoid discovering duplicate sequences. Note that the total order has no effect on the final result generated by SPM. Thus any total order relation can be used. In this work, the order $\prec$ is simply defined on AAs as the lexicographical order, that is $A \prec C \prec D \prec E \prec F \prec G \prec H \prec I \prec K \prec L \prec M \prec N \prec P \prec Q \prec R \prec S \prec T \prec V \prec X \prec Y$. SPM algorithms either employ a depth-first search or a breadth-first search. Moreover, SPM algorithms utilize the Apriori (also known as anti-monotonicity) property to avoid exploring the whole search space. This property basically states that for S_α and S_β, if S_α is a subsequence of S_β, then the support of S_β should be equal or less than that of S_α. For instance, if a sequence that contains ($\langle L \rangle$) has a support of 3, then the sequence $\langle LV \rangle$ cannot have a support greater than 3. This property helps in reducing the search space as it proves that the extensions of an infrequent sequence are also infrequent. Thus, they cannot be considered as sequential patterns. For example, if $minsup = 4$, finding the extensions of $\langle L \rangle$ is unnecessary as they are all infrequent. SPM algorithms differ from each other in the following aspects:

1. Whether a depth-first search or breadth-first is used,
2. Whether a vertical database representation or horizontal database representation is used, and internal data structures,
3. How the support is computed for those patterns that satisfy the $minsup$ constraint.

SPAM [58], TKS [59] and CM-SPAM [60] are some efficient SPM algorithms.

The CM-SPAM [60] algorithm is an improved version of SPAM. To discover all the sequential patterns, CM-SPAM uses a depth-first search and a vertical database representation. CM-SPAM utilizes the CMAP (Co-occurrence MAP) data structure, that saves information about item co-occurrences, to reduce the search space. However, in CM-SPAM, setting the $minsup$ value on a new dataset is not intuitive. This is because a large value of $minsup$ may result in discovering no patterns or missing important patterns. On the other hand, a small value of $minsup$ may generate large patterns, most of which are redundant. To overcome this, the TKS (Top-k Sequential) algorithm was proposed. A user can find a certain number of patterns by using the TKS algorithm through the parameter k. In TKS, various strategies are used for the search space reduction.

In genome sequences, sometimes it is also interesting to discover those sets of AAs that occur frequently without taking into account the sequential ordering. *Frequent itemset mining* (FIM) [61], which is a special case of SPM, can be used for this purpose. Let AAS represents an *amino acids set*, such that $AAS \subseteq AAs$. For a $SGSC$ and a $minsup$ threshold ($minsup > 0$), the *support* of AAS, represented as $sup(AAS)$, in $SGSC$ is the total occurrence of sequences containing AAs from AAS: $sup(AAS) = |\{S|\exists x \in S\ \forall x \in AAS\}|$. The task of FIM in $SGSC$ is to enumerate all *frequent amino acids sets*. If $sup(AAS) \geq minsup$, then AAS is frequent. For example, the amino acids set $\{N, C, E, F\}$ in Table 3 is frequent as the four genome sequences contain this AAS.

For FIM, Apriori [62] is the first algorithm and is arguably the most famous. In large databases, Apriori can discover frequent itemsets and proceeds by finding those common items that are extendable to larger itemsets that appear frequently. Itemsets (AAS here) discovered with Apriori can also be used to derive association rules (relationships) among items. Some fast and memory efficient FIM algorithms are now present in the literature.

In this paper, *sequential rules* are also studied in genome sequences. The motivation to discover these rules is that frequent sequential patterns provide frequent subsequences of AAs. However, some discovered frequent sequential patterns may be redundant or spurious because these are found without checking the probability or confidence that some AAs follow others. Therefore, sequential frequent patterns may be misleading in some cases. Algorithms developed for mining sequential rules find patterns by taking into account both their support and confidence [63]. Thus for a genome sequence, a sequential rule $Y \rightarrow Z$ represents a relationship between two AAs $Y, Z \subseteq AAS$, such that $Y \cap Z = \emptyset$ and $Y, Z \neq \emptyset$. A rule $r : Y \rightarrow Z$ means that if items of Y appear in a sequence, items of Z will appear afterward in the same sequence.

An AA set Y is contained in a sequence S_α (written as $Y \sqsubseteq S_\alpha$) if an only if $Y \subseteq \bigcup_{i=1}^{n}\{\alpha_i\}$. A rule $r : Y \rightarrow Z$ is present or contained in S_α ($r \sqsubseteq S_\alpha$) if and only if there exists an integer k such that $1 \leq k < n$, $Y \subseteq \bigcup_{i=1}^{k}\{\alpha_i\}$ and $Z \subseteq \cup_{i=k+1}^{n}\{\alpha_i\}$. The support and confidence of a rule r in a $SGSC$ are calculated as:

$$sup_{SGSC}(r) = \frac{|\{S|r \sqsubseteq S \wedge S \in SGSC\}|}{|SGSC|}$$

$$conf_{SGSC}(r) = \frac{|\{S|r \sqsubseteq S \wedge S \in SGSC\}|}{|\{S|X \sqsubseteq S \wedge S \in SGSC\}|}$$

For a $SGSC$, a $minsup > 0$ (user-defined minimum support) and a $minconf \in [0, 1]$ (minimum confidence threshold), if $sup_{SGSC}(r) \geq minsup$ then the rule r is a *frequent sequential rule*. On the other hand, r is a *valid sequential rule* iff it is frequent and $conf_{SGSC}(r) \geq minconf$. Mining sequential rules task deals with discovering all the valid sequential rules in a corpus.

ERMiner (Equivalence class based sequential Rule Miner) [63] is a representative sequential rule mining algorithm that uses a vertical database representation and depends on the equivalence classes of rules concept that have the same antecedent and consequent. Two operations are used in ERMiner, namely left and right merges, for the generation of larger rules from smaller rules. The Sparse Count Matrix (SCM) technique is used to reduce the search space. In [63], it was shown that ERMiner performed better than several other previous sequential rule mining algorithms.

In this study, another task performed is to use sequence prediction models to see if the arrangement of AAs is predictable in SARS-CoV-2 genome sequences. For this, several popular models are used to examine their performance and compare their results to see which model performs well. The models used in this paper include CPT+ [64], CPT [65], DG [66], AKOM [67], Mark1 [68], TDAG [69] and LZ78 [70]. DG (Dependency Graph) [66] is a light-weight Markov based model that is used to perform sequence predictions. DG takes a set of training sequences as input and computes the probabilities that each symbol is followed by each symbol. However, in DG, only the last symbol is used for the prediction of the next symbol. The AKOM (All-k Order Markov) [67] model overcomes the limitation of DG by considering the last k symbols in the prediction. The value of k is set by the user. On the other hand, in Mark1, only the current symbol is used for the prediction of the next symbol. In LZ78 [70] and TDAG (Transition Directed Acyclic Graph) [69], data compression techniques are used in the prediction.

The Compact Prediction Tree (CPT) and its improved version CPT+ consider more than one symbol and their different orderings for prediction. Their main drawback is that they require a large memory. For prediction, CPT+ takes as input a set of training sequences and generates three data structures: (1) a prediction tree, (2) a lookup table and (3) an inverted index. These three structures are constructed incrementally and during the training process, each sequence is considered one by one. For S_α containing n elements, the suffix of S_α with the size y ($1 \leq y \leq n$) is defined as $P_y(S_\alpha) = \langle \alpha_{n-y+1}, \alpha_{n-y+2}, ..., \alpha_n \rangle$. For S_α, predicting the next AA is performed by discovering sequences that are similar to $P_y(S_\alpha)$. The discovered sequences can be in any order. CPT+ also uses each sequence *consequent* that is similar to S_α in the prediction. Let S_β be another sequence that is similar to S_α. With respect to S_α, the consequent of S_β is the longest subsequence $\langle \beta_v, \beta_{v+1}, ..., \beta_m \rangle$ of S_β such that $\bigcup_{k=1}^{v-1}\{\beta_k\} \subseteq P_y(S_\alpha)$ and $1 \leq v \leq m$. The count table (CT) data structure stores each AA that is found in the consequent of a similar sequence of S_α. In last, the AA with the highest support in the CT is returned by CPT+ as prediction.

The SPMF data mining library [57], developed in Java, implements more than 230 data mining algorithms. In this work, we use SPMF since it implements the aforementioned SPM and sequence prediction algorithms.

5 Experiments and results

The results obtained by using the AF methods discussed in Section 4.1 and SPM algorithms in Section 4.2 on SARS-CoV-2 genome sequences are presented in this section.

Table 4 presents the statistics about the collected genome sequences. Through GenBank, one can download genome sequences in three forms: (1) *nucleotide*, (2) *coding region*, and (3) *protein*. In this work, the genome sequences in *protein form* are used. Note that NC_045512 is the *RefSeq* (reference sequence) for SARS-CoV-2 in NCBI. This sequence was released by the Public Health Clinical Center and School of Public Health in Shanghai, China [1]. A ninth generation Intel Celeron processor laptop with 16 GB RAM is used to perform the experiments.

5.1 Results for AF methods

We first provide the results obtained by using AF methods offered by Alfree, followed by the results obtained by using AF methods offered by CAFE on SARS-CoV-2 genome sequences.

5.1.1 Alfree results

We ran Alfree on the corpus that contains the sequences listed in Table 4 to find frequent AA sets. Some frequent AAs that are discovered by Alfree in the *RefSeq* (NC_045512) and in all the sequences are listed in Table 5. It is interesting to find that the extracted frequent AAs in one genome (NC_045512) and all genome sequences are almost the same. For example, the most frequent AA was *Leucine* (*L*) in NC_045512 and in all genome sequences. Same is the case for other frequent amino acid patterns. There are a total of 25,650 AAs in NC_045512 and approximately 9.62% of them are *L*, followed by Valine (*V*) with a share of approximately 8.26%. On the other hand, there are 87,935 AAs in all considered genome sequences and approximately 9.60% of them are *L*, followed by *V* with a share of approximately 8.13%. It is

Table 4 Characteristics of SARS-CoV-2 genome sequences

Accession Number	Date of Release	Location	Date of Collection
NC_045512	2020-01-13	China	2019-12
MW052550	2020-11-03	South Korea	2020-07-07
MW192918	2020-10-31	Gabon	2020-03-14
MW173089	2020-10-26	USA	2020-04-25
MW165491	2020-10-24	Iran	2020-04
MW161041	2020-10-23	Russia	2020-06-04
MW092768	2020-10-12	Sweden	2020-02-25
MW040503	2020-09-26	Venezuela	2020-05-22
MT843234	2020-08-28	Italy	2019-12-18
MT750057	2020-07-13	USA	2020-06-17
MT750058	2020-07-13	USA	2020-06-09
MT291827	2020-04-06	China	2019-12-30
MT291828	2020-04-06	China	2019-12-30

Table 5 Extracted frequent AAs

NC_045512				**All sequences**			
Patterns	**Occurrence**	**Patterns**	**Occurrence**	**Patterns**	**Occurrence**	**Patterns**	**Occurrence**
L	2468	LL	236	L	8394	LL	826
V	2121	VL	231	V	7152	VL	782
T	1935	LK	219	T	6630	LK	702
A	1763	VV	210	A	5989	VV	689
S	1705	LA	185	S	5961	LA	601
K	1548	LLS	32	G	5245	LLL	113
G	1516	LLL	31	K	5150	LLS	97
N	1382	GVV	28	N	4833	GVV	91
D	1322	TTT	28	I	4537	ALL	90
I	1286	VLL	27	D	4469	VLL	90
F	1265	LKTL	12	F	4420	LKTL	36
E	1260	LLSV	12	E	4189	LLSV	36
Y	1173	TTTL	12	Y	3984	TTTL	36
P	996	VVTT	12	P	3479	VVTT	36
Q	906	EGSV	10	Q	3244	NLLL	34
R	856	EVVLK	8	R	2975	EVVLK	24
C	798	LAKAL	8	C	2707	LAKAL	24
M	585	LEGSV	8	M	1894	LEGSV	24
H	484	LLSVL	8	H	1606	LLSVL	24
W	281	LSEQL	8	W	957	LSEQL	24

important to point here that Alfree was fast and generated the frequent AAs in seconds.

Various distance measure in Alfree are used next to compute the pairwise (dis) similarities between the SARS-CoV-2 genome sequences. Figure 5 shows the heatmaps for the two word-based methods (d^{E} and d^{Canberra}), the information theory-based method (d^{NCD}) and d^{W}. It is observed that the four measures produced different heatmaps for the same sequences. Interestingly, the heatmap of $d^{Canberra}$ was most similar to the heatmap of d^{NCD} as compared to d^{E} and d^{W}.

The calculated distance by various word and information theory-based AF methods on two genome sequences (MT750057 and MT750058) is shown in Table 6 for comparison. The calculated (dis)similarity by d^{S} and d^{E} (both are Euclidean distance-based measures) was very different, whereas the results for d^{Eseq1} and d^{Eseq2} were the same. The two measures $d^{Minkowski}$ and d^{S} generated the same results. The Angle metrics measures (d^{EVOL1} and d^{EVOL2}) also generated the same results. The performance of five absolute-based metrics measures (d$^{abs\_mean}$, d$^{abs\_Manhattan}$, d$^{abs\_Bray-Curtis}$, d$^{abs\_Canberra}$ and d^{Chebyshev}) was different from each other. Similarly, the performance of three Boolean vectors-based measures (d$^{Sorenson\_dice}$, d^{Jaccard} and d^{Hamming}) was different from each other. The Kullback-Leibler divergence (D^{KL}) uses entropy-based measure to compare genome sequences. The two information theory-based methods (d^{BBC} and d^{NCD}) performed almost similarly. In summary, we find that different AF methods can be compared easily and efficiently in Alfree.

The constructed phylogenetic tree for the genome sequences of SARS-CoV-2 in Table 4 with three measures (d^{E}, d^{Canberra} and d^{NCD}) is shown in Fig. 6 as dendogram. The consensus tree that summarizes the agreement between various AF methods is also shown in Fig. 6. For an organism or group of organisms, a phylogeny (also known as phylogenetic tree) or evolutionary tree diagrammatically illustrates the relationship and evolutionary history. Phylogeny relationships can provide very important information related to shared ancestry. Here, the phylogenetic tree is built for 24 AF methods. Out of 24, 21 are word-based methods, 2 are information theory-based methods and 1 is a hybrid method. In the consensus tree, the range [0, 1] is used to represent the support values of all nodes. The four trees in Fig. 6 describe how strains of a virus (SARS-CoV-2 in this work) are connected with each other and how they have evolved.

Note that the *k-mer* word frequencies-based method (d^{FCGR}) and the three graphical representation-based methods (d^{2DSV}, d^{2DMV} and d^{2DNV}) work on DNA sequences in Alfree. Next, we provide the results obtained by using various AF methods in CAFE.

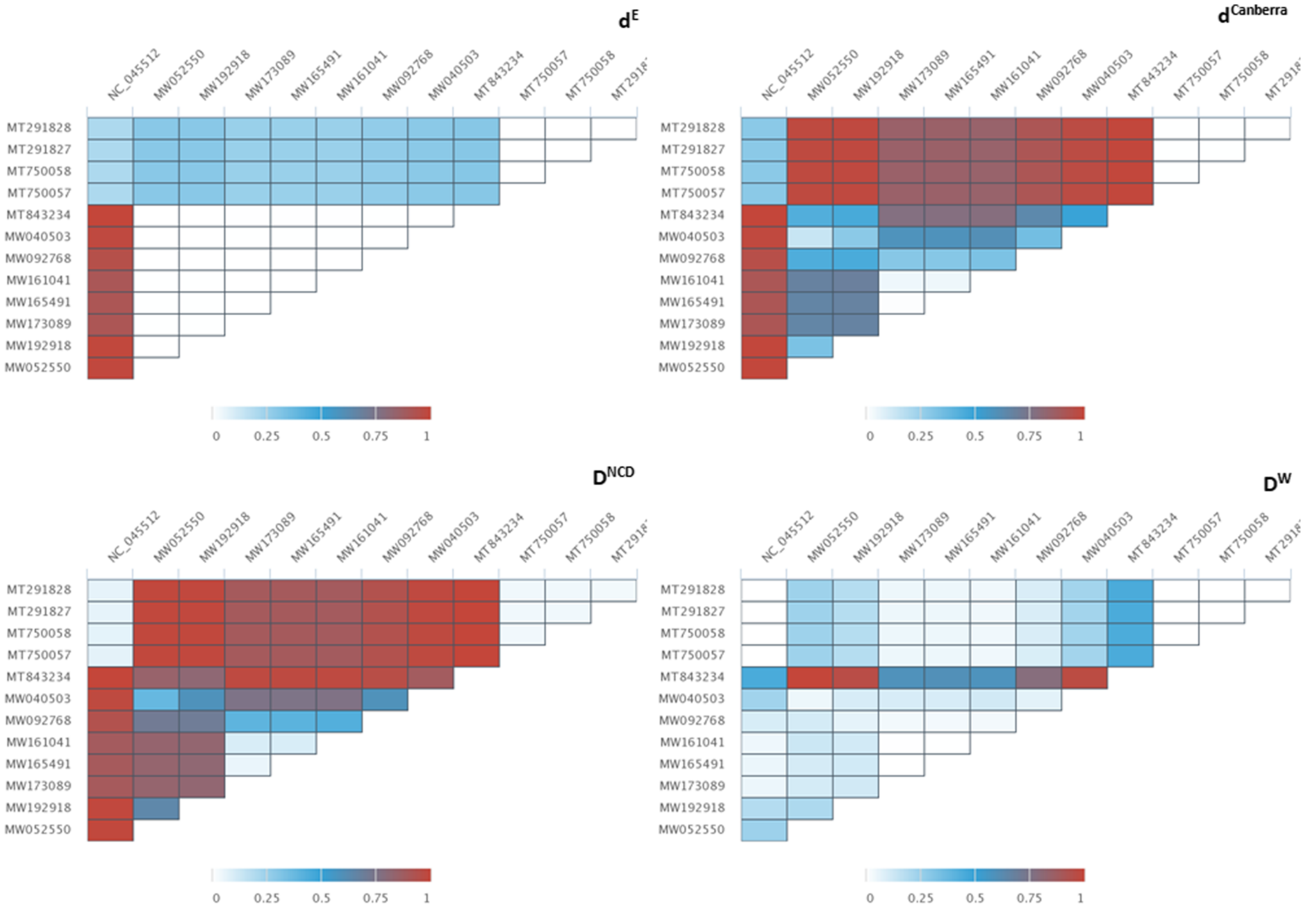


Fig. 5 Calculated distance among genome sequences by using d^{E}, $d^{Canberra}$, d^{NCD} and d^{W}

Table 6 Measured distance for MT750057 and MT750058

Measures	AD*	ND**	Measure	AD	ND
		MT750057, MT750058			
d^{E}	132	0.001	d^{S}	11.489	0.025
d^{Eseq1}	0.009	0.001	d^{Eseq2}	0.009	0.001
d^{Minkowski}	11.489	0.025	d$^{abs\_mean}$	0.015	0.003
d^{BBC}	0.003	0.005	d^{Manhattan}	72	0.003
d$^{Bray\_Curtis}$	0.002	0.003	d^{Canberra}	16.499	0.004
d^{EVOL1}	0.0005	0.001	d^{EVOL2}	0.0005	0.001
d^{FFP}	0.00016	0.001	d^{Google}	0.0025	0.003
d^{LCC}	0.0014	0.003	d^{Chebyshev}	2	0.063
d^{KL}	3.67	0.000	d$^{Sorenson\_Dice}$	0.0007	0.001
d^{Jaccard}	0.0014	0.001	d^{Hamming}	7	0.002
d^{RTD}	0.004	0.004	d^{CV}	0.0004	0.001
d^{NCD}	0.035	0.036			

*AD: The actual distance, **ND: The normalized distance

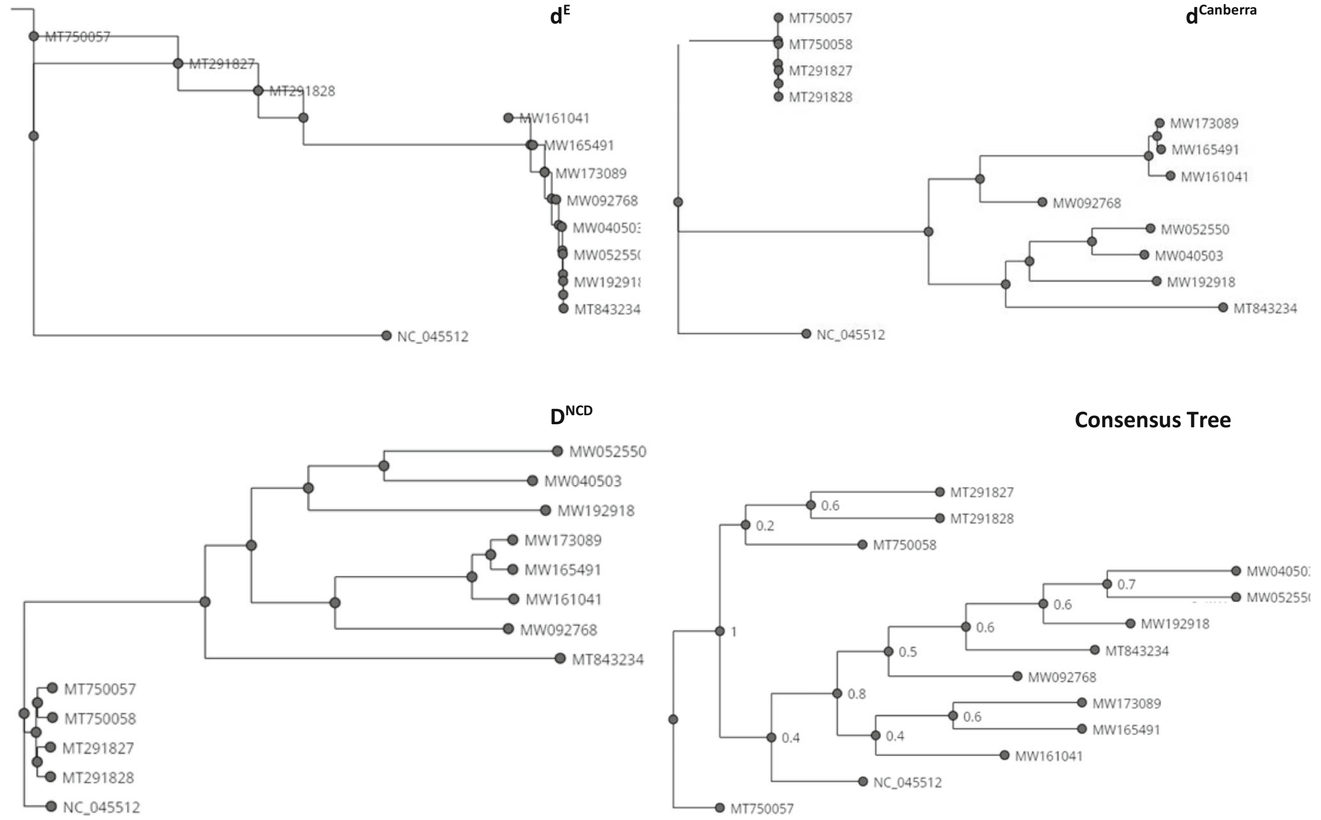


Fig. 6 Phylogeny for SARS-CoV-2 strains

5.1.2 CAFE results

The heatmaps obtained for different SARS-CoV-2 strains in CAFE are shown in Fig. 7. The heatmaps are obtained by using four measures (Manhattan, Pearson, Hamming and d_2^S). The value of K is set to 4 for all four measures, whereas Markov order 3 is used for the d_2^S measures that is based on background adjusted *k-mer* counts. It can be seen that the *k-mer*-based AF dissimilarity measures (Manhattan and Pearson) generated the same heatmaps whereas the heatmap for the measure based on background adjusted *k-mer* counts (d_s^S) is more similar to the heatmap for the Hamming distance measure that is based on absence/presence of *k-mers* as compared to the heatmaps for Manhattan and Pearson measures. The phylogenetic trees for different SARS-CoV-2 strains in CAFE are shown in Fig. 8. The trees are obtained by using four measures with the same parameter for K and markov order. Interestingly, the trees for three measures (Manhattan, Hamming and d_2^S) are very similar to each other.

Compared to Alfree, CAFE does not provide the feature of counting the frequent patterns of nucleotides and AAs. Moreover, one can not run more than one measure at the same time, and can only get the results for one measure at a time. Alfree provides the implementation for the Kullback-Leibler (KL) divergence, whereas CAFE implements a more smooth and symmetric version of KL divergence called the Jensen-Shannon (JS) divergence. Computation wise, we found that CAFE speed is about the same as Alfree. Obviously, Alfree takes more time in the case where more than one measure is used to analyze strains of SARS-CoV-2. CAFE provides results for principal component analysis (PCoA) and network analysis that are discussed next.

Figure 9 shows the 2-dimensional projections obtained by using the PCoA with four measures. PCoA is a statistical method that summarizes the (dis)similarity among SARS-CoV-2 strains in a low-dimensional, Euclidean space. From the figure, we can see that four measures generated different projections. Figure 10 shows the network analysis using four measures for strains of SARS-CoV-2 with respect to the 10% quantile of the edges with smallest distance as weight. Again, the results for four measures were different from each other.

NCBI uses alignment-based method called Basic Local Alignment Search Tool (BLAST) [71] for genome analysis and building a phylogenetic tree. A study [72] used the Natural Vector AF method for the analysis of the phylogeny among SARS-CoV-2 and human coronaviruses. Through the distance measured among SARS-CoV-2 and coronavirus genomes residing in animals, it was established that

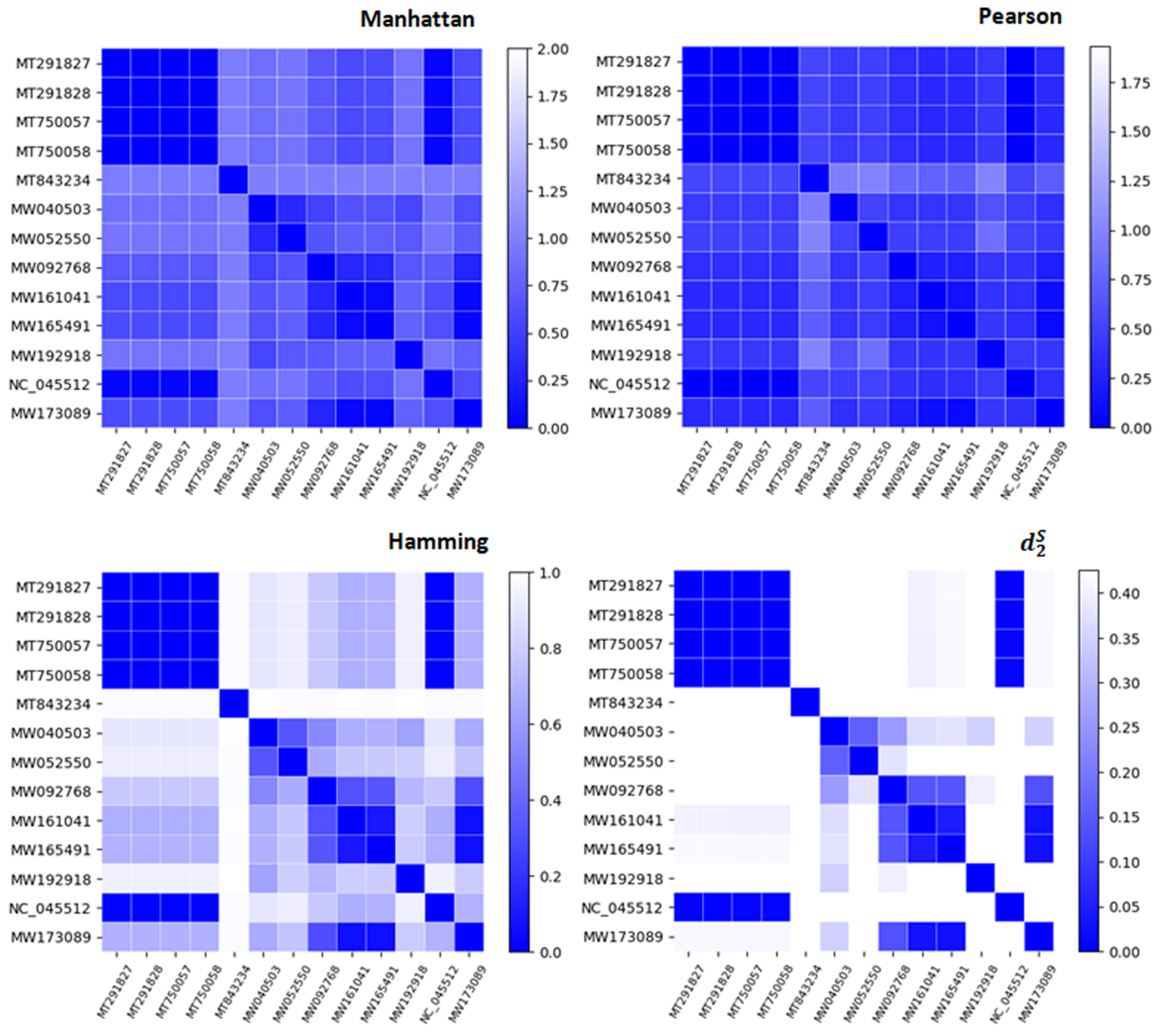


Fig. 7 Distance between genome sequences calculated using Manhattan, Pearson, Hamming and d_S^S measures

SARS-CoV-2 was likely transmitted from bats to pangolins to humans. We also found that AF methods can efficiently analyze and compare genome sequences. Alfree provides certain features that can be used to extract frequent patterns from nucleotides and AAs and to build consensus phylogenetic tree that is not available in CAFE. Whereas, CAFE provides the PCoA analysis and network analysis features that are not available in Alfree.

5.2 Results for SPM

This section presents the obtained results by using SPM algorithms on the corpus.

5.2.1 Frequent AA sets

The frequent AA sets are discovered in the corpus first by applying the Apriori algorithm. Apriori takes a *minsup* threshold and a corpus as input and generates the frequent AA sets as output. Table 7 lists the extracted AA sets by Apriori in one genome sequence (MT291828) for different *minsup* values. We found that for the *minsup* in the range of 70% - 100%, Apriori produced only 19 frequent patterns. By decreasing *minsup* to 10%, Apriori produced 35 patterns and for *minsup* of 1%, Apriori produced 283 patterns. Note that the amino acid Tryptophan (*W*) is not included as its occurred only 157 times. It occurred when *minsup* is set to 60%. Similar to the results obtained for frequent AAs with

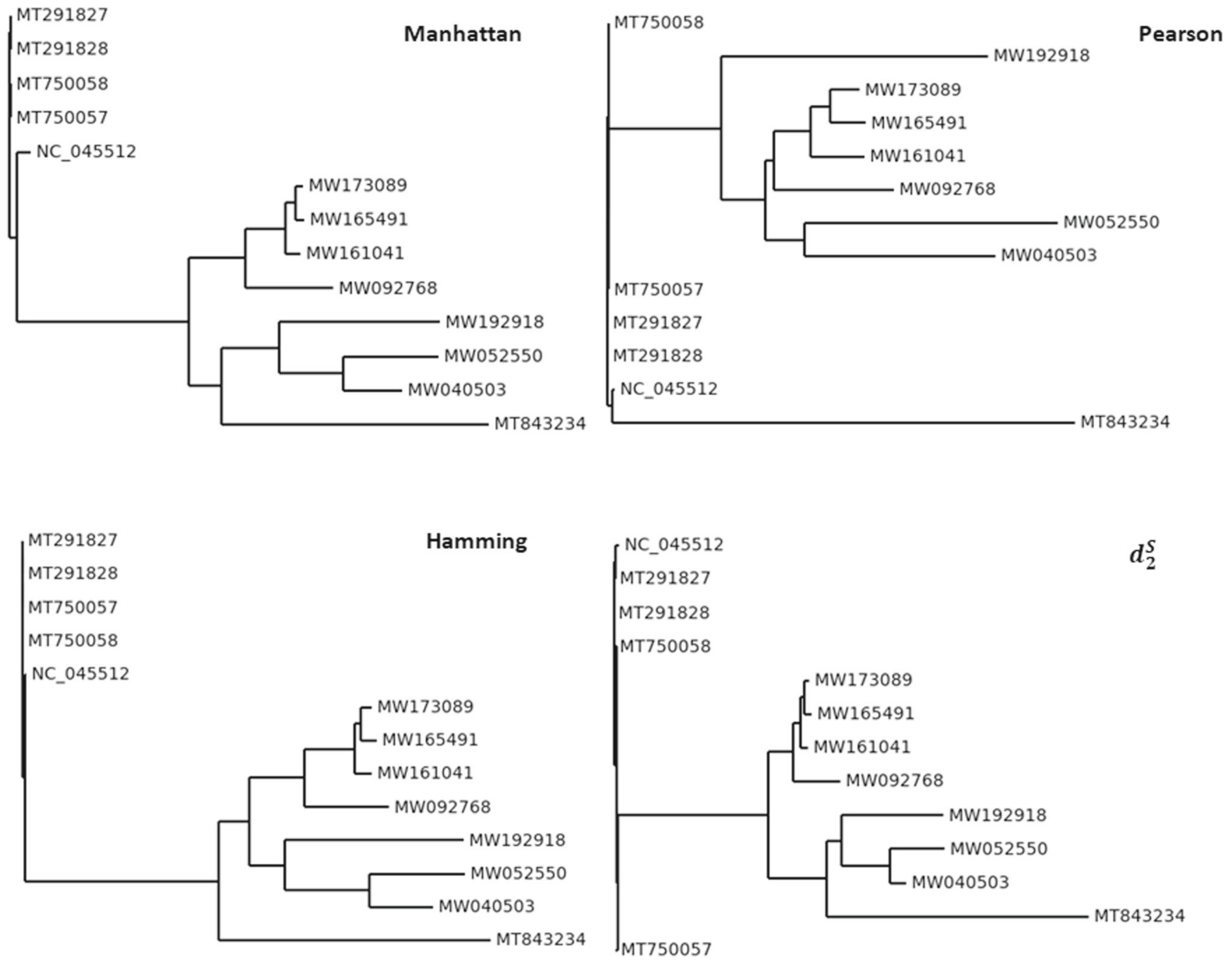


Fig. 8 Phylogeny for SARS-CoV-2 strains using different measures

Alfree in Table 5, it can be seen in Table 7 that in MT291828, *L* occurred the most frequently, followed by *V*.

For biologists, the frequent AA sets may not be interesting because they are unordered and the Apriori algorithm does not make sure that an AA from the AA set occur contiguously in a genome sequence. This means that an AA set can be considered as occurring in a genome sequence if all its AAs occurs in it. However, the AAs are separate from each other in the sequence through some other sub-patterns. These sub-patterns are called *gaps*.

Next, the results for SPM algorithms are presented that overcome the aforementioned limitations. Thus they find and reveal more interesting and meaningful patterns.

5.2.2 Frequent sequential patterns

SPM algorithms are used on genome sequences in the corpus to find hidden sequential patterns among AAs. First, the CM-SPAM algorithm, that requires a $minsup$ threshold and a corpus, is executed.

Some frequent patterns for AA that are discovered in MT291828 by using CM-SPAM are listed in Table 8. The first three patterns on the left side are found for a $minsup$ of at least 95% of the lines in the genome sequence (MT291828). For example, *SL* has a support of 232 and occurred in approximately 230 lines in MT291828. Similarly, the first three patterns on the right side occurred in at least 70% of lines in MT291828.

The mining process to discover frequent AAs was fast in terms of computation time and memory usage. For various $minsup$ threshold values, the performance of CM-SPAM is provided in Table 9. It can be see that CM-SPAM finds more frequent patterns by decreasing $minsup$, while the runtime and the memory usage increases.

Next, the TKS algorithm was applied for mining the top-k sequential AA patterns. TKS takes *k* (the total number of patterns that a user needs) and a corpus as input. It returns

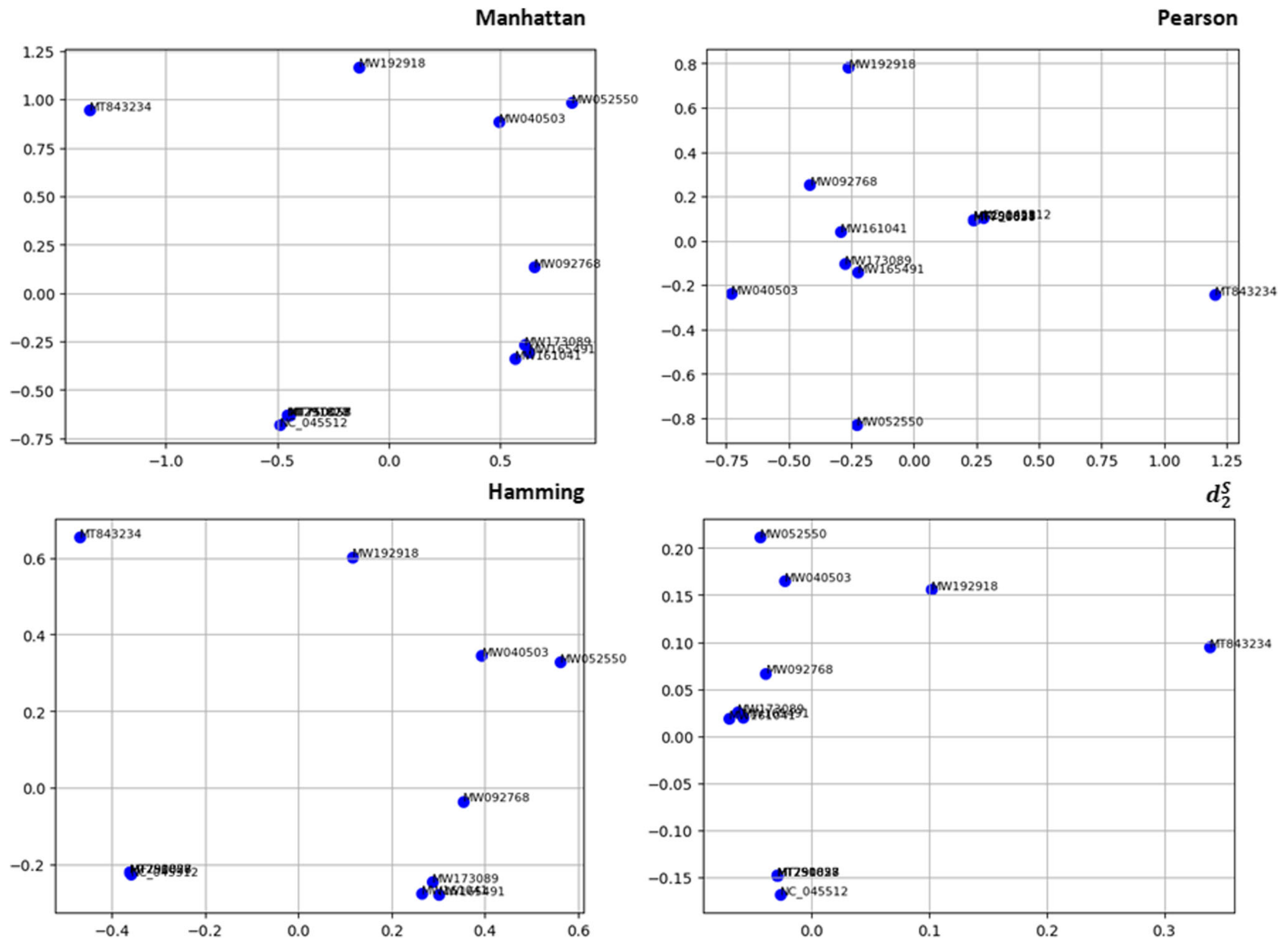


Fig. 9 PCoA of SARS-CoV-2 strains using four measures

the top-k most frequent sequential patterns as output. The parameter k is used in TKS in place of minusp because:

1. Selecting the proper *minsup* to find the desired useful patterns affects the performance of SPM algorithms. For example, we already observed that with a decrease in *minsup*, CM-SPAM generated more frequent patterns. Most of the patterns may be redundant and its hard for users to look for specific patterns in a file that contain thousands of patterns.
2. It is hard and time consuming to fine-tune the *minsup* value to find enough but not too many patterns.

Thus, the parameter k overcomes these limitations by putting a bound on the total discovered patterns by TKS. Some top frequent AAs patterns discovered in MT291828 by the TKS algorithm of different lengths are shown in Table 10. We can see that frequent patterns for AAs discovered by CM-SPAM and TKS are similar to each other. Moreover, we observe that AAs that occurred in most sequences appear frequently in frequent AA patterns.

5.2.3 Sequential rules

Next, sequential rules in the genome sequences are studied. Some sequential rules discovered by the ERMiner algorithm in one genome sequence (MT291828) are shown in Fig. 11 that indicate a strong relationships among AAs. Both the *minconf* and *minsup* were set to 90%. Thus, those rules were discovered by ERMiner that have confidence of 90%, at least. In a sequential rule $Y \rightarrow Z$ with a 90% confidence, AAs set in Z follows AAs set in Y no more than 90% of the times when Y occurs in a sequence. For *minconf* and *minsup* of 90%, 28 sequential rules were generated by ERMiner in total. In Fig. 11, the values above and below an arrow is for the support and confidence respectively. The first rule $S \rightarrow A$ shows that the amino acid *Alanine* (A) follows the amino acid *Serine* (S) 93.2% of the times with a support

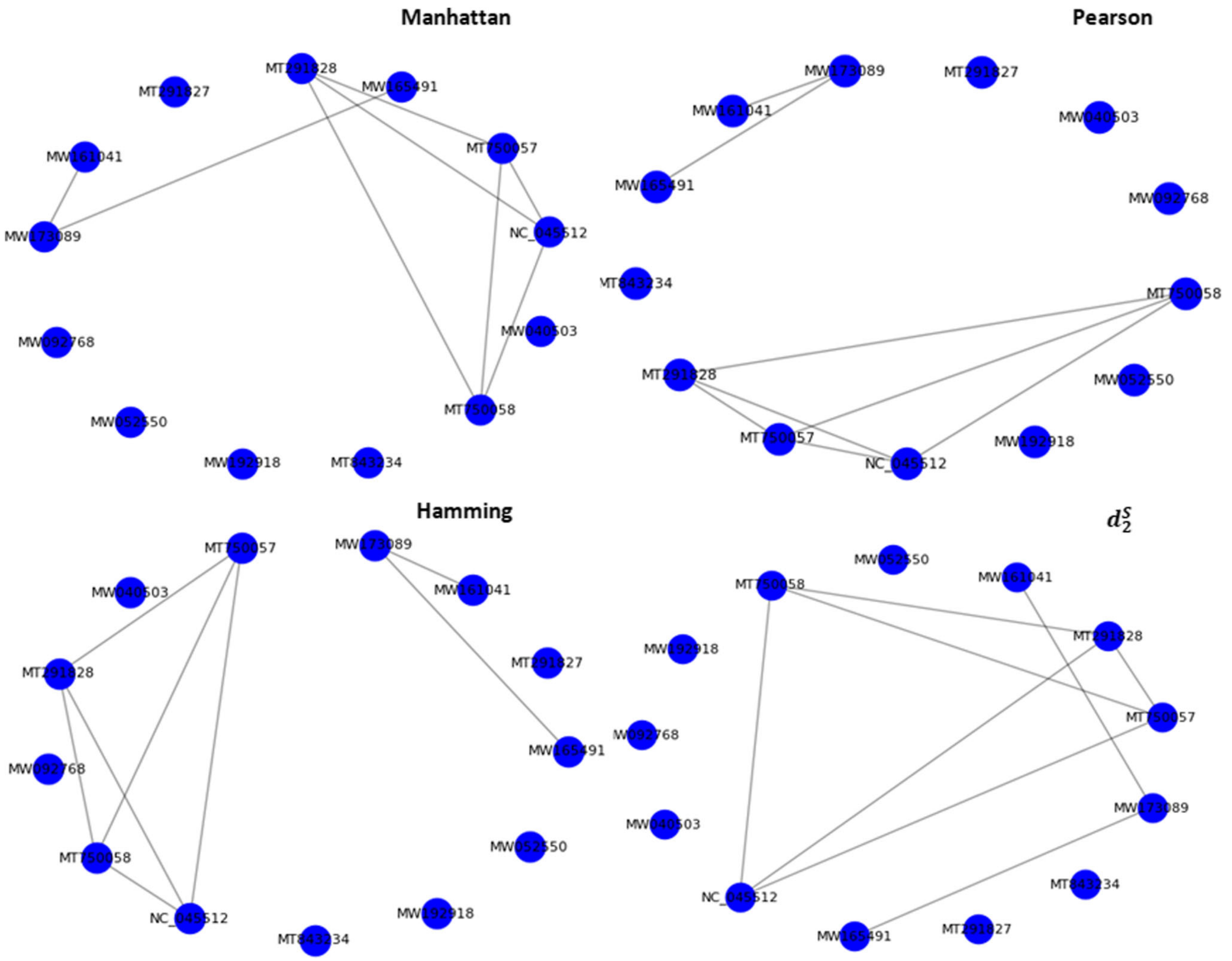


Fig. 10 Network display of SARS-CoV-2 strains using four measures

of 222. We found that the total number of AAs in a genome sequence has an effect on the performance of SPM algorithms and the discovered frequent patterns and sequential rules.

5.2.4 Amino acid (AA) prediction

AA(s) prediction in SARS-CoV-2 genome sequence is performed to investigate the predictability of a genome sequence. To predict the next AA and their patterns, several models are compared. AAs and their frequent patterns in a genome sequence are used to train each model. After training, the next AA and their patterns in a genome sequence are predicted by using a prediction model. Predicting the next AA and their pattern(s) is based on the scores computed by the prediction model for each AA.

The performance of seven prediction models (LZ78, AKOM, DG, CPT+, CPT, TDAG and Mark1) are compared. 10-fold cross-validation is used for model training and testing. In machine learning, cross-validation is a statistical method that is used for characterizing the performance of models. Through cross-validation, one can evaluate and access the generalization of independent dataset over the results of statistical analysis provided by the model. In the k-fold cross-validation, the dataset is divided randomly into k equal sized subdatasets. In k subdatasets, one subdataset is used as the validation set to test the model and the remaining $k-1$ sub-datasets are used to train the model. This process is used k times which means that each subdataset is used once as the validation set. The average of k results provides the single estimation for the overall result . 10-fold cross-validation is used here to make sure that low variance is achieved during each run. Table 11 provides the details about the dataset containing only one genome sequence (MT291828).

The results of the prediction models are interpreted and evaluated with three measures: (1) *success*: if the model correctly predicts, (2) *failure*: if the model incorrectly predicts, and (3) *no match* if the model is unable to predict. For all prediction, the three measures (*success*, *failure* and *no match*)

Table 7 Discovered frequent AA sets

Pattern(s)	Support	Min. Support	Pattern(s)	Support	Min. Support
A	967	100%	VY	135	50%
C	434	100%	TY	74	30%
D	733	100%	SY	65	20%
E	681	100%	TV	50	20%
F	708	100%	SV	52	15%
G	840	100%	SVY	40	15%
H	264	100%	TVY	44	15%
I	728	100%	PY	34	10%
K	838	100%	LY	35	10%
L	1365	100%	PVY	25	10%
M	312	100%	LVY	23	5%
N	765	100%	RVY	19	5%
P	558	100%	VWY	18	5%
Q	516	100%	STVY	13	5%
R	481	100%	NVY	20	4%
S	955	100%	LPVY	9	3%
T	1063	100%	PTVY	9	3%
V	1152	100%	PSTVY	5	2%
Y	643	100%	GPTVY	4	1%

are expressed as percentage. Among these measures, the *accuracy* is generally considered the most important measure for model comparison as it shows the ability of a model to perform better predictions. Besides, the training and testing times for the models were measured in seconds.

Results for the prediction models are shown in Table 12 for MT291828. This table also includes the results for a baseline model, called *Random* that randomly predicts the next AA in a genome sequence.

We found that CPT+ and CPT achieved the highest accuracy (100%), followed by AKOM, TDAG and LZ78. On the other hand, DG, Mark1 and Random generate the same results with the highest number of failures. Both CPT+ and CPT performed well as they take into account not only many symbols but also their different orderings for prediction.

Table 8 Frequent AAs extracted by using the CM-SPAM algorithm

Pattern	Support	Min. Sup	Pattern	Support	Min. Sup
LL	233	95%	AVLL	182	70%
SL	232	95%	GLLL	179	70%
VL	234	95%	VNLL	176	70%
AL	224	90%	LLALL	147	60%
VLT	219	90%	TLVVL	147	60%
VT	228	90%	VSVLL	149	60%
VLL	224	90%	ALAVL	123	50%
AAL	196	80%	AVLLLL	100	40%
VVT	200	80%	AAALLL	74	30%
VSL	217	80%	YYVTGV	52	20%

Table 9 CM-SPAM performance for varying $minsup$

Min. Sup %	Time (Sec)	Patterns	Memory (Mb)	Min. Sup.
95%	0.26	8	41.40	231
80%	0.123	231	32.85	195
60%	1.5	3,583	125.786	146
40%	15.9	47,411	125.78	98
20%	272	136,984,0	125.31	49

We also found two more interesting observations: (1) *No Match* is always zero which means that all the models were able to do the predictions, and (2) models training and testing times were very low and remained quite similar.

Other than CPT+ and CPT, the accuracy of six models to predict AA in genome sequences was comparatively low. The main reason for this is the fact that genome sequences contain 20 distinct AAs and their distribution is unequal (or not uniform). The prediction models has one main limitation. They can be used to predict only one AA in the genome sequence.

6 Mutation

This section presents the approach to analyze SARS-CoV-2 genome sequences in *protein form* for mutations.

Taking prior work [12] as starting point, the focus here is on mutation analysis in SARS-CoV-2 genome sequences in *protein form* and the pseudocode is presented in Algorithm 1. Algorithm 1 takes as input two genome sequences of SARS-CoV-2 (GS_1 and GS_2) and compares the AAs in the two sequences line by line. The line numbers and the locations in line where AAs are different are stored in a set called $Diff$,

Table 10 Frequent AAs extracted by using TKS algorithm

Pattern	Length	Support	Pattern	Length	Support
VLT	3	219	NVTLP	5	85
SL	2	232	TLLTGL	6	85
GS	2	217	AVLLLL	6	100
AAL	3	196	YYVTGV	6	52
VNLL	4	176	GVVLLLV	7	72
ALLL	4	182	LLLLVTL	7	75
AL	2	224	VLL	3	224
LLTLV	5	144	TLVVL	5	147

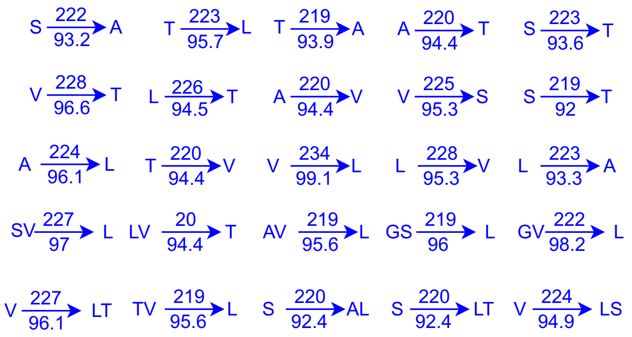


Fig. 11 Discovered sequential rules by using the ERMiner algorithm

that also stores the changed AAs. The following formula is used to calculate the mutation rate (MR):

$$MR = \frac{TM}{TAA} \times 100 \qquad (1)$$

where TM represents the number of changes that have occurred and TAA represents the total number of AAs.

Algorithm 1 Mutation Analysis of SARS-CoV-2 genome sequences in protein form

Input: Genome sequences in protein form (GS_1, GS_2)
Output: Locations with changed AAs in GS_1, GS_2 and the mutation rate

```
$Diff \leftarrow \emptyset$;
$TL \leftarrow$ total lines in $GS_1$, $GS_2$;                ▷ $len(GS_1) = len(GS_2)$
$TM, TAA \leftarrow 0$
for $k \leftarrow 1$ to $TL$ do
    for $i \leftarrow 1$ to length($TL$) do
        if $GS_1(i) \neq GS_2(i)$ then
            $Diff \leftarrow k, i, GS_1(i), GS_2(i)$;
            $TM \leftarrow TM + 1$
        end if
        $TAA \leftarrow TAA + 1$
    end for
    $TAA \leftarrow TAA + 1$
end for
$MR \leftarrow \frac{TM}{TAA} \times 100$
return $Diff, MR$
```

This algorithm is implemented in Python. The algorithm is first run on two genome sequences (MT750057 and MT750058) from Table 4. The algorithm returns the locations and lines number where the AAs in MT750057 and MT750058 are changed (shown in Table 13). The column four and column eight in Table 13 provide the information for the changed AAs. For example, the first entry ($S \rightarrow P$) in column four shows that the amino acid *Serine* (S) (in MT750057) was replaced by the amino acid *Proline* (P) (in MT750058). The last amino acid (Valine (V) → Leucine (L)) change in Table 13 occurred in the S protein of MT750058. Whereas other changes occurred in *ORF1ab* and *ORF1a* polyproteins. It was observed that the strain MT750058 has 12 locations where AAs are changed (listed in Table13) than MT750057. In a previous study [12], we found 8 changes in MT750057 and MT750058 genome sequences that were downloaded from NCBI in *nucleotide form*.

Table 11 Statistics for the dataset used for sequence prediction

Parameter	Value
Sequences number	240
Distinct items	21
ID of Itemsets	25
Distinct items per sequence	18.53
Each item occurrence	3.23
Size of the corpus size in MB	960.057

For two genome sequence (MT750057 and MT750058), the computed mutation rate is 0.0848% Similarly, for MT291827 and MT291828, the computed mutation rate is 0.000%, which means that no AAs were changed. In our prior study [12] on analyzing the SARS-CoV-2 genome sequences in *nucleotide* form, we found that one nucleotide A (in MT291827) was changed to G (in MT291828). Because of this, the codon AAA (in MT291827) was changed to AAG (in MT291828). However, both AAA and AAG make the same amino acid called *Lysine*. Besides, note that China genome sequences (MT291827 and MT291828) were reported and submitted earlier to NCBI than USA genome sequences (MT750057 and MT750058). Moreover, the China genome sequences belong to the same city. Whereas the USA genome sequences belong to different cities. Through initial results obtained in this section, different mutation rates were observed for the genome sequences of SARS-CoV-2. Moreover, the results indicate an increase in the mutation rate with the passage of time. In summary, the developed algorithms in this paper and in prior work [12] for mutation analysis can be used to analyze:

- The point mutation and mutation rate for genome sequences in both *nucelotide* and *protein* forms.
- Whether the changes in nucleotides indeed changes the AAs that makes the proteins or the changes in nucleotides only change the codon that encodes the same amino acid.
- There are variations between mutations from place to place. Strains of SARS-CoV-2 from different locations (places) can be analyzed to investigate whether they coexist with each other or not. A similar study [73] found

Table 12 Prediction models accuracy

Models	DG	TDAG	CPT+	CPT	Mark1	AKOM	LZ78	Random
Success	0	57.917	100	100	0	57.917	4.583	0
Failure	100	42.083	0	0	100	42.083	95.417	100
No Match	0	0	0	0	0	0	0	0
Train Time	0.005	0.045	0.026	0.002	0.001	0.038	0.004	–
Test Time	0	0	0.001	0.011	0	0	0.001	–

that SARS-CoV-2 strains from Asia, North America and Europe might coexist.

.

The algorithm for the mutation analysis is compared with some of the recent (published in last two years) approaches. The work [38] investigated the nucleotide mutation rate and codon mutation rate in genomes of SARS-CoV-2. The computed mutation rates in [38] for genomes from China, Australia, USA and rest of the world were 4.775%, 7.3%, 6.65% and 6.725% respectively. The highest computed codon mutation rate in [38] was 0.12%. Moreover, the work [37] used Needleman-Wunsch (NW) and Smith-Waterman (SW) algorithms to analyze two genome sequences (MT750057 and MZ359831) against the *RefSeq*. They found that MT750057 and MZ359831 has seven and nine nucleotide differences, respectively, when compared to *RefSeq*. Both NW and SW scores for (MT750057, *RefSeq*) and (MZ359831, *RefSeq*) are 18308 and 18303 respectively. The works [37, 38] compared the genomes with the *RefSeq* for mutation analysis. The codons in [38] were converted into positive integers (from 1 to 64) on the basis of their index number. Our previous work [12] investigated the mutation rate in SARS-CoV-2 genome sequences in *nucleotide form*. The computed mutation rate in [12] for genomes from China and USA were 0.0268% and 0.0003% respectively. Whereas the computed amino acid mutation rate in this work for genomes from China is 0.848%. In [12] and in this work, the genomes are compared with each other and not with the the reference genome sequence.

The approaches for mutation analysis presented here and in [12] have two limitations. First, it is required that the length of both genome sequences be the same. Note that that the AF methods and SPM algorithms do not suffer from this length limitation and both work well on genome sequences of varying length. Second, the input to the mutation analysis approaches is the whole genome sequences without the metadata that provides information for proteins and ORFs. In the future, we intend to work on these two limitation. The main aim is the development of a generic approach that has the ability to analyze genome sequences in *nucleotide*, *coding region* and *protein* forms and can handle genome sequences of (un)equal lengths.

7 Conclusion

Genome sequences of SARS-CoV-2 in *protein form*, taken from NCBI's GenBank, have been compared and analyzed with AF and SPM methods to: (1) extract frequent AAs in the sequences, (2) find the (dis)similarity between sequences by using various distance measures and their performance comparison, (3) construct phylogeny using various AF methods for sequences, (4) find frequent patterns of AAs and the sequential relationships between such patterns and (5) predict the next amino acid in a sequence. Moreover, for mutation analysis, an algorithm was proposed that can find the location(s) in genome sequences where the AAs are changed to compute the rate of mutation. We found that AF and SPM provide efficient frameworks to compare and analyze SARS-CoV-2 genome sequences. The analysis and learning approaches are not limited to the SARS-CoV-2 virus and can be used to compare and analyze other human viruses too.

There are many interesting future work opportunities such as:

Table 13 Results for mutation analysis of AAs in MT750057 and MT750058

Line	Location	Position	Change	Line	Location	Position	Change
23	1	1,321	S → P	31	3	1,803	S → P
37	54	2,214	S → L	43	4	2,524	F → L
46	9	2,709	I → V	108	51	6,471	H → Q
142	1	8,416	S → P	150	3	8,899	S → P
156	54	9,310	S → L	162	4	9,620	F → L
165	9	9,805	I → V	212	42	12,623	V → L

- Applying descriptive pattern mining sub-tasks such as contrast pattern (set) mining, also called emerging pattern mining [74], on the SARS-CoV-2 genome sequences. This will allow to find contrasting (or emerging) patterns (trends) in sequences that show a clear difference between two classes or disjoint features.
- Using pattern mining and deep learning to predict the codon pairs and codon families in the genome sequences of SARS-CoV-2.
- Improving the mutation analysis approach in Section 6 to make it more general. For example, proposing strategies to overcome the limitation of the sequence length and adding the gene information. Moreover, the mutation detection technique can be extended for the comparison of a new genome sequence with a dataset of SARS-CoV-2 genomic sequences.

Acknowledgements This work was supported by Natural Science Foundation of Guangdong Province (2023A1515011667) and Basic Research Foundations of Shenzhen (JCYJ20210324093609026)

Data Availability The code for Algorithm 1 in Python and the genome sequences used in the experiments are available at: https://github.com/saqibdola/SPM-MA4GSA/tree/master/MAP.

Declarations

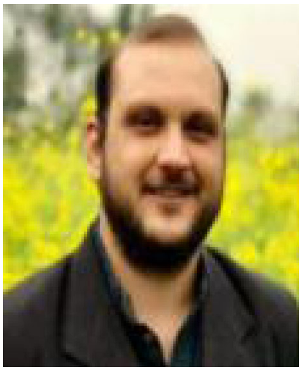
Conflict of interest Authors declare no conflict on interest.

References

1. Wu F et al (2020) A new coronavirus associated with human respiratory disease in China. Nature 579:265–269
2. Coronaviridae Study Group of the International Committee on Taxonomy of Viruses (2020) The species Severe acute respiratory syndrome-related coronavirus: Classifying 2019-nCoV and naming it SARS-CoV-2. Nat Microbiol 5:536–544
3. Mount DM (2004) Bioinformatics: Sequence and Genome Analysis, 2nd edn. Cold Spring Harbor Laboratory Press
4. Aggarwal C, Bhuiyan M, Hasan M (2014) Frequent pattern mining algorithms: A survey. In: Frequent Pattern Mining, Springer
5. Zielezinski A et al (2017) Alignment-free sequence comparison: Benefits, applications, and tools. Genome Biol 18:186
6. Vinga S (2014) Information theory applications for biological sequence analysis. Brief Bioninf 15(3):376–389
7. Vinga S, Almeida J (2003) Alignment-free sequence comparison-A review. Bioinformatics 19:513–523
8. Zielezinski A et al (2019) Benchmarking of alignment-free sequence comparison methods. Genome Biol 20:144
9. Fournier-Viger P et al (2017) A survey of sequential pattern mining. Data Sci Patt Recog 1:54–77
10. Karim MR et al (2013) An efficient approach to mining maximal contiguous frequent patterns from large DNA sequence databases. Genomics Informat 10(1):51–57
11. Kawade DR, Oza KS (2013) Exploration of DNA sequences using pattern mining. J Biomed Informa 2:144–148
12. Nawaz MS, Fournier-Viger P, Shojaee A, Fujita H (2021) Using artificial intelligence techniques for COVID-19 genome analysis. Appl Intell 51(5):3086–3103
13. Ni L et al (2020) Mining the local dependency itemset in a products network. ACM Trans Manage Infor Syst 11 (1): 3:1-3:31
14. Mustafa RU et al (2017) Early detection of controversial urdu speeches from social media. Data Scie Patt Recogn 1(2):26–42
15. Pokou YJM, Fournier-Viger P, Moghrabi C (2016) Authorship attribution using small sets of frequent part-of-speech skip-grams. In: Proceedings of FLAIRS, pp. 86-91
16. Nawaz MS, Fournier-Viger P, Zhang J (2020) Proof learning in PVS with utility pattern mining. IEEE Access 8:119806–119818
17. Nawaz MS, Sun M, Fournier-Viger P (2019). Proof guidance in PVS with sequential pattern mining. In: Proceedings of FSEN, pp. 45-60
18. Schweizer D et al (2015) Using consumer behavior data to reduce energy consumption in smarthomes: Applying machine learning to save energy without lowering comfort of inhabitants. In: Proceedings of ICMLA, pp. 1123-1129
19. Nawaz MS et al (2022) MalSPM: Metamorphic malware behavior analysis and classification using sequential pattern mining. Computers & Security 118:102741
20. Fournier-Viger P, Gueniche T, Tseng VS (2012). Using partially-ordered sequential rules to generate more accurate sequence prediction. In: Proceedings of ADMA, pp. 431-442
21. Nawaz MS et al (2021) COVID-19 genome analysis using alignment-free methods. In: Proceedings of IEA AIE, pp. 316-328
22. Rondo HM et al (2021) Pathogenesis, symptomatology, and transmission of SARS-CoV-2 through analysis of viral Genomics and structure. mSystems 6(5): e00095-21
23. Nawaz MS, Fournier-Viger, P, He Y (2022) S-PDB: Analysis and classification of SARS-CoV-2 Spike protein structures. In: Proceedings of BIBM, pp. 2259-2265
24. Khailany RA, Safdar M, Ozaslanc M (2020) Genomic characterization of a novel SARS-CoV-2. Gene Reports 19:100682
25. Shu J-J (2017) A new integrated symmetrical table for genetic codes. Biosystems 151:21–26
26. Mohamadou Y, Halidou A, Kapen PT (2020) A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19. Appl Intell 50:3913–3925
27. Nayak J et al (2021) Intelligent system for COVID-19 prognosis: A state-of-the-art survey. Appl Intell 51:2908–2938
28. Alyasseri Z et al (2021) Review on COVID-19 diagnosis models based on machine learning and deep learning approaches. Expert Systems e12759
29. Lalmuanawma S, Hussain J, Chhakchhuak L (2020) Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. Chaos Solito 139:110059
30. Chen J, See JC (2020) Artificial intelligence for COVID-19: Rapid review. J Med Internet Res 22:e21476
31. Rasheed J et al (2021) COVID-19 in the age of artificial intelligence: A comprehensive review. Interdiscip Sci Comput Life Sci 13:153–175
32. Shi F et al (2021) Review of artificial intelligence techniques in imaging data acquisition, segmenta-tion and diagnosis for COVID-19. IEEE Rev Biomed Engg 21:4–15
33. Driggs D et al (2021) Machine Learning for COVID-19 diagnosis and prognostication: Lessons for amplifying the signal while reducing the noise. Radiology: Artificial Intelligence 3(4): e210011
34. Roberts M et al (2021) Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans. Nat Mach Intell 3:199–217
35. Wynants L et al (2020) Prediction models for diagnosis and prognosis of COVID-19: Systematic review and critical appraisal. BMJ 369:m1328

36. Noor S et al (2020) Analysis of public reactions to the novel coronavirus (COVID-19) outbreak on Twitter. Kybernetes 50(5):1633–1653
37. Heng JW, Juwono FH, Reine R (2021) Using optimal sequencing algorithms for COVID-19 case study. In: Proceedings GECOST, pp. 1-4
38. Pathan RK, Biswas M, Khandaker MU (2020) Time series prediction of COVID19 by mutation rate analysis using recurrent neural network-based LSTM model. Chaos Solit 138:110018
39. Zelenova M (2021) Analysis of 329,942 SARS-CoV-2 records retrieved from GISAID database. Comput Biol Med 139:104981
40. Kali K (2021) The lag in SARS-CoV-2 genome submissions to GISAID. Nat Biotechnol 39:1058–1060
41. Arslan H (2021) Machine learning methods for COVID-19 prediction using human genomic data. Proceedings 74(1), 20
42. Arslan H, Arslan H (2021) A new COVID-19 detection method from human genome sequences using CpG island features and KNN classifier. Int J Eng Sci Technol 24(4):839–847
43. Arslan H (2021) COVID-19 prediction based on genome similarity of human SARS-CoV-2 and bat SARS-CoV-like coronavirus. Comput Ind Eng 161:107666
44. Lopez-Rincon et al (2021) Classification and specific primer design for accurate detection of SARS-CoV-2 using deep learning. Scient Rep 11:947
45. Naeem SM (2021) A diagnostic genomic signal processing (GSP)-based system for automatic feature analysis and detection of COVID-19. Brief Bioinf 22(2):1197–1205
46. Randhawa GS et al (2020) Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study. PLoS One 15(4):e0232391
47. Ahmed I, Jeon G (2021) Enabling artificial intelligence for genome sequence analysis of COVID-19 and alike viruses. Interdiscip Sci 6:1–16
48. Ren J et al (2018) Alignment free sequence analysis and applications. Annu Rev Biomed Sci 1:93–114
49. Bonham-Carter O et al (2014) Alignment-free genetic sequence comparisons: A review of recent approaches by word analysis. Brief Bioinf 15(6):890–905
50. Song J et al (2014) New developments of alignment-free sequence comparison: measures, statistics and next-generation sequencing. Brief Bioinf 15(3):343–353
51. Lu YY et al (2017) CAFE: aCcelerated Alignment-FrEe sequence analysis. Nucleic Acids Res 45(Web Server issue): W554-W559
52. Frigessi A, Heidergott B (2011) Markov Chains. In: Lovric M (ed) International Encyclopedia of Statistical Science. Springer
53. Otu HH, Sayood KA (2003) A new sequence distance measure for phylogenetic tree construction. Bioinformatics 19(1):2122–2130
54. Li M et al (2004) The similarity metric. IEEE Trans Infor Theory 50(12):3250–64
55. Giancarlo R, Rombo SE, Utro F (2014) Compressive biological sequence analysis and archival in the era of high-throughput sequencing technologies. Brief Bioinf 15(3):390–406
56. Sayers EW et al (2019) Genbank. Nucleic Acids Res 48(D1):D84–D86
57. Fournier-Viger P et al (2016). The SPMF open-source data mining library version 2. In: Proceedings ECML PKDD, pp. 36-40
58. Ayres J (2002). Sequential pattern mining using a bitmap representation. In: Proceedings KDD, pp. 429-435
59. Fournier-Viger P et al (2013) TKS: Efficient mining of top-k sequential patterns. In: Proceedings of Advanced Data Mining and Applications (ADMA), pp. 109-120
60. Fournier-Viger P (2014). Fast vertical mining of sequential patterns using co-occurrence information. In: Proceedings of PAKDD, pp. 40-52
61. Aggarwal CC, Han J (2014) Frequent Pattern Mining. Springer
62. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large databases. In: Proceedings VLDB, pp. 487-499
63. Fournier-Viger P (2014). ERMiner: Sequential rule mining using equivalence classes. In: Proceedings of IDA, pp. 108-119
64. Gueniche T et al (2015) CPT+: Decreasing the time/space complexity of the compact prediction tree. In: Proceedings of PAKDD, pp. 625-636
65. Gueniche T, Fournier-Viger P, Tseng VS (2013). Compact prediction tree: A lossless model for accurate sequence prediction. In: Proceedings of AADMA, pp. 177-188
66. Padmanabhan VN, Mogul JC (1996) Using predictive prefetching to improve world wide web latency. Comp Comm Rev 26:22–36
67. Pitkow J, Pirolli P (1999) Mining longest repeating subsequence to predict world wide web surfing. In: Proceedings of USENIX Symposium on Internet Technologies and Systems, pp. 13-25
68. Deshpande M, Karypis G (2004) Selective markov models for predicting web page accesses. ACM Trans. Inter. Techn. 4:163–184
69. Laird P, Saul R (1994) Discrete sequence prediction and its applications. Machine Learning 15:43–68
70. Ziv J, Lempel A (1978) Compression of individual sequences via variable-rate coding. IEEE Trans. Infor. Theory. 24:530–536
71. Altschul SF et al (1990) Basic local alignment search tool. J. Molec. Biolo. 215(3):403–410
72. Dong et al (2020) Analysis of the hosts and transmission paths of SARS-CoV-2 in the COVID-19 outbreak. Genes 11(6):637
73. Pachetti M et al (2020) Emerging SARS-CoV-2 mutation hot spots include a novel RNA-dependent-RNA polymerase variant. J. Transl. Medi. 18:179
74. Ventura S, Luna JM (2018) Supervised Descriptive Pattern Mining. Springer

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

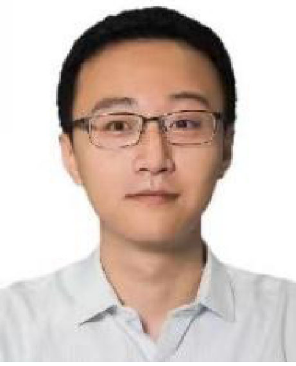


M. Saqib Nawaz received the B.S. degree in Computer Systems Engineering from the University of Engineering and Technology, Peshawar, Pakistan in 2011, the M.S. degree in Computer Science from the University of Sargodha, Pakistan in 2014, and the Ph.D. degree from Peking University, Beijing, China, in 2019. He worked as a postdoctoral fellow at Harbin Institute of Technology (Shenzhen) from September 2019 to January 2022. He is currently working as associate researcher at Shenzhen University, China. His research interests include bioinformatics, pattern mining, formal methods and the use of machine learning and data mining in software engineering.

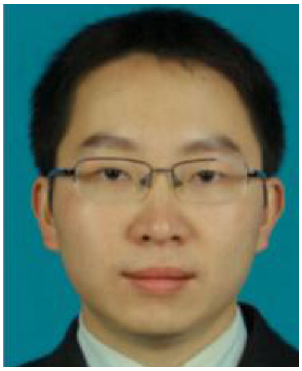
Philippe Fournier-Viger (Ph.D) is distinguished professor at Shenzhen University, China. He has published more than 330 research papers related to data mining, intelligent systems and applications, which have received more than 12,000 citations (H-Index 56). He was associate editor-in-chief of the Applied Intelligence journal and editor-in-chief of Data Science and Pattern Recognition. He is the founder of the popular SPMF data mining library. He is a co-founder of the UDML, PMDB and MLiSE series workshop at the ICDM, PKDD, DASFAA and KDD conferences. His interests are data mining, algorithm design, pattern mining, sequence mining, big data and applications.

Memoona Aslam received her Bachelor's degree in Zoology from University of Punjab, Lahore, Pakistan in 2019. She is currently pursuing her Master's degree in Life Health and Environment from Institute of Advanced Study, Shenzhen, China. Her research interests include Bioinformatics, Genomes, Molecular Docking and Molecular Dynamics Simulations.

Wenjin Li obtained the PhD in Computational Biology from Shanghai Institutes for Biological Sciences of CAS and now an assistant professor at the Institute for Advanced Study of Shenzhen University. His research focuses on the development of computational methods for enhanced sampling, free energy calculations, and the identification of reaction coordinates. These approaches are then applied to study enzymatic reactions, drug discovery and protein engineering, by combining MM and QM/MM molecular dynamics.



Yulin He was born in 1982. He received the Ph.D. degree from Hebei University, China, in 2014. From 2011 to 2014, he has served as a Research Assistant with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, China. From 2014 to 2017, he worked as a Post-doctoral Fellow in the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He is currently a Research Associate with Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China. His main research interests include big data approximate computing technologies, multi-sample statistical analysis theories and methods, and data mining/machine learning algorithms and their applications. He has published over 100+ research papers in ACM Transactions, CAAI Transactions, IEEE Transactions, Elsevier, Springer Journals and PAKDD, IJCNN, CEC, DASFAA conferences. Dr. He is an ACM member, CAAI member, CCF member, IEEE member, and the Editorial Review Board members of several international journals.



Xinzheng Niu was born in 1978. He received the Ph.D. degree from University of Electronic Science and Technology of China in 2008. From 2007 to 2009, he has had the privilege of serving as a Visiting Scholar at the University of Illinois at Chicago. From 2012 to 2021, he was an Associate Professor at the School of Computer Science and Engineering, University of Electronic Science and Technology of China. He is currently a professor-level Senior engineer at the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His main research interests include data analysis, data mining and image recognition. He has published over 70+ research papers in IEEE Transactions, Information Sciences and other international and domestic academic journals and conferences. He is an IEEE member, ACM member, senior member of the Computer Society and the reviewer of several journals both domestically and internationally.

Authors and Affiliations

M. Saqib Nawaz[1] · Philippe Fournier-Viger[1] · Memoona Aslam[2] · Wenjin Li[2] · Yulin He[3] · Xinzheng Niu[4]

M. Saqib Nawaz
msaqibnawaz@szu.edu.cn

Memoona Aslam
memunasaqib93@gmail.com

Wenjin Li
liwenjin@szu.edu.cn

Yulin He
yulinhe@gml.ac.cn

Xinzheng Niu
xinzhengniu@uestc.edu.cn

[1] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

[2] Institute for Advanced Study, Shenzhen University, Shenzhen, China

[3] Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), 518107 Shenzhen, China

[4] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China