PSAC-PDB: Analysis and Classification of Protein Structures

M. Saqib Nawaz[a], Philippe Fournier-Viger[a,*], Yulin He[a,b], Qin Zhang[a]

[a] *College of Computer Science and Software Engineering, Shenzhen University, China*
[b] *Guangdong Laboratory of Artificial Intelligence & Digital Economy (SZ), Shenzhen, China*

Abstract

This paper presents a novel framework, called PSAC-PDB, for analyzing and classifying protein structures from the Protein Data Bank (PDB). PSAC-PDB first finds, analyze and identifies protein structures in PDB that are similar to a protein structure of interest using a protein structure comparison tool. Second, the amino acids (AA) sequences of identified protein structures (obtained from PDB), their aligned amino acids (AAA) and aligned secondary structure elements (ASSE) (obtained by structural alignment), and frequent AA (FAA) patterns (discovered by sequential pattern mining), are used for the reliable detection/classification of protein structures. Eleven classifiers are used and their performance is compared using six evaluation metrics. Results show that three classifiers perform well on overall, and that FAA patterns can be used to efficiently classify protein structures in place of providing the whole AA sequences, AAA or ASSE. Furthermore, better classification results are obtained using AAA of protein structures rather than AA sequences. PSAC-PDB also performed better than state-of-the-art approaches for SARS-CoV-2 genome sequences classification.

Keywords: Protein structures, SARS-CoV-2, Spike, SPM, PDB, DALI, Classification.

1. Introduction

Proteins, the building block of all living organisms, are one of the fundamental macromolecule families that can perform more than one function and govern biology, by being extensively involved in almost all biological mechanisms [1]. Proteins contribute to our understanding of human health and therapies for particular diseases. Proteins contain long chains of amino acids (called protein sequences) that are connected and fold into three-dimensional (3D) structures.

*Corresponding author
Email addresses: msaqibnawaz@szu.edu.cn (M. Saqib Nawaz), philfv@szu.edu.cn (Philippe Fournier-Viger), yulinhe@gml.ac.cn (Yulin He), qinzhang@szu.edu.cn (Qin Zhang)

Structural biology experimental techniques [2] such as Cryo-EM (electron microscopy) and X-Ray crystallography are commonly used to view, in atomic resolution, how proteins assemble, function and interact. Experimental methods for structure determination of proteins are slow, costly and cannot be used to determine the structure of some proteins. Thus, computational methods are quite desirable for the prediction and classification of protein structure from sequence information.

For optimal protein structure alignment and comparison, various computational tools have been proposed and developed in the last two decades such as DALI [3], GrAfSS [4], FATCAT [5], MICAN-SQ [6], MADOKA [7], DeepAlign [8], Cassert [9], iPBA [10], Fr-TM-Align [11], TM-ALign [12], FAST [13] and PDBeFold [14]. Among publicly available structure comparison algorithms, DALI is the most popular and best known for comparing protein structures from the PDB database [4, 5, 7]. The SCOP database [15] and its extensions [16, 17, 18, 19] use a combination of manual curation and automated methods to provide detailed information for structural and evolutionary relationships among all known protein structures. Protein structural classification and prediction is an important research topic and the recent use of deep learning methods [20, 21] to accurately predict protein structures has greatly increased the scope of comparative structural studies that was constrained in the past by the labor and cost required in experimental structure acquisition.

Some recent machine and deep learning-based studies focused on the classification and detection of diseases by analyzing genome sequences obtained from online databases for genomic data such as GenBank [22] and GISAID [23]. For example, some studies [24, 25, 26] took advantage of CpG (or CG)-based features to classify genomes of the SARS-CoV-2 virus. Representative genomic sequences were discovered by Lopez-Rincon et al. [27] by combining a deep learning method with explainable AI techniques. Naeem et al. [28] developed a classification system that extracted features from genome sequences using the discrete Fourier transform and seven moment invariants. The classification method of Randhawa et al. [29] uses an intrinsic genomic signature with a machine learning-based alignment-free (AF) method. Ahmed and Jeon [30] classified genome sequences of four viruses (SARS-CoV-1, SARS-CoV-2, MERS and Ebola) by using ML algorithms. Singh et al. [31] used biomarkers, that were extracted from the genome sequences of coronaviruses on the basis of three-base periodicity, for the classification of SARS-CoV-2 from other coronaviruses. Most of these studies focused on virus genome sequences and finding important features in them that are then used for classification. To the best of our knowledge, no study has been published on the classification of protein structures based on pattern mining, particularly for harmful viruses, in Protein Data Bank (PDB) [32].

The main aim of this study is to investigate how structural alignment method such as DALI [3] and sequential pattern mining (SPM) [33], a special case of structured data mining, can be used for the analysis and classification/detection of protein structures given their abundance in the PDB. More specifically, based on the analysis of amino acids (AA) sequences of protein structures and their secondary structure elements (SSE), a novel approach called PSAC-PDB is proposed to:

1. Find and determine the protein structures in PDB that are similar to a protein structure of interest. Identified similar protein structures are then analyzed and compared by using a structural alignment technique to find aligned AA and SSE. Moreover, SPM is used to analyze the AA sequences of similar protein structures to discover frequent AA and their frequent patterns.
2. Detect and classify the protein structures. Four kinds of classification are carried out that are based on the (1) amino acids (AA) sequences, (2) aligned AA (AAA) sequences, (3) aligned secondary structure elements (ASSE) sequences and (4) frequent AA (FAA) patterns. Three text-based and eight integer-based classifiers are used for classification and their efficacy is accessed with six evaluation metrics.

As a case study, the proposed PSAC-PDB approach was applied to investigate SARS-COV-2 [34] by considering its Spike (S) protein [35, 36] as the protein structure of interest. We discovered that using SPM to first find frequent AA patterns in the AA sequences of protein structures and using these patterns yield better classification performance than using only the AA sequences, AAA and ASSE. Additionally, classification with AAA outperformed classification with AA. Using ASSE for classification produced poor results. Three classifiers (two text-based and one integer-based) performed well, on overall. Moreover, text-based classifiers required more time for training and testing than integer-based classifiers. The performance of PSAC-PDB was also compared with the state-of-the-art approaches for SARS-CoV-2 classification and detection from genome sequences and results show that PSAC-PDB outperforms these approaches. We believe that the developed framework and obtained results will benefit the research community in general, and specifically crystallographers and biochemists.

The rest of the paper is divided into three sections: Section 2 presents the proposed PSAC-PDB method, as well as the protein structure types and datasets. Section 3 presents and discusses the obtained results. Finally, Section 4 draws a conclusion and outlines some future research directions.

2. PSAC-PDB

The proposed PSAC-PDB method (Figure 1) for the analysis and classification of protein structures in PDB consists of three main steps:

1. *Similar protein structures identification*: The first step is to find and identify protein structures in PDB that are similar to a protein structure of interest. Here our interest is in the S protein structures of SARS-CoV-2. Similar protein structures are found using DALI via the PDB90 search strategy.

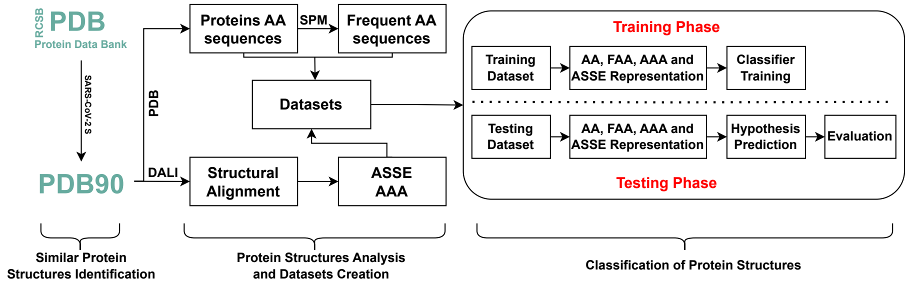

Figure 1: Schematic of the PSAC-PDB method for the analysis and classification of proteins structures in PDB. PSAC-PDB is applied in three main steps: (1) Identification of similar protein structures in PDB, (2) Analysis of similar protein structures by using structural alignment and SPM, and datasets creation, and (3) Classification of protein structures by training various classifiers using the created datasets.

2. *Protein structures analysis and Datasets Development*: Three main activities are performed in this step: (1) Pairwise structural alignment of AA and SSE of protein structures. (2) Protein structures AA sequences are analyzed using SPM to discover common AA and the frequent AA (FAA) patterns. (3) AA, AAA, ASSE and FAA are stored in four datasets.
3. *Protein structures classification*: The datasets for AA, FAA, AAA and ASSE sequences are used for the classification of protein structures. The classification task is composed of two main parts: (1) The training phase contains two phases, which are AA, FAA, AAA and ASSE representation, and classifier training, and are performed sequentially. (2) The testing phase consists of three phases, AA, FAA, AAA and ASSE representation, hypothesis prediction, and evaluation.

PDB now contains over 0.202 million structures, in which 0.173 million are X-Ray structures and 14,511 are Cryo-EM structures[1]. Manually finding and selecting structures for analysis and classification is difficult and time consuming. Thus, Step 1 helps in finding protein structures in PDB. The AA sequences of selected protein structures are then analyzed in Step 2 and the developed datasets for AA, AAA, ASSE and FAA are then used for classification and detection. Next, we provide some details for DALI and SPM.

DALI optimizes a set of one-to one correspondences between two protein (sub)structures, say A and B, that maximizes the DALI score:

$$DALI_{AB} = \sum_{i=1}^{LALI} \sum_{j=1}^{LALI} \left(\theta - \frac{2|d_{ij}^{A} - d_{ij}^{B}|}{d_{ij}^{A} + d_{ij}^{B}} \right) e^{-\left(\frac{d_{ij}^{A} + d_{ij}^{B}}{2D}\right)^{2}}$$

where $LALI$ is the number of aligned residue pairs, $\theta = 0.2$, $D = 20$ A and d_{ij}^{A}, d_{ij}^{B} are intra-molecular Cα–Cα distances in structures A and B respectively.

[1] rcsb.org/stats

For random pairwise comparison, DALI$_{AB}$ score increases with the number of residues in the compared proteins. In DALI, Z-score is used to describe the statistical significance of DALI$_{AB}$:

$$Z_{AB} = \frac{DALI_{AB} - m(L)}{\alpha(L)}$$

where $L = \sqrt{L_A L_B}$ is the geometric mean length of structures A and B. The relation between the mean score m, standard deviation σ and L was derived empirically from a large set of random pairs of structures. Fitting a polynomial gives the following approximation:

$$m(L) = \begin{cases} 7.95 + 0.71L - 2.59E^{-4}L^3 - 1.92E^{-6}L^3 & \text{if } L \leq 400 \\ m(400) + L - 400, & \text{if } L > 400 \end{cases}$$

For the standard deviation, an empirical estimate is $\sigma(L) = 0.5 \times$ m(L). The Z-score is computed for every possible pair of domains and the highest value is reported as the Z-score of the protein pair. Thus DALI's Z-score is an optimized similarity score defined as the sum of equivalent residue-wise Cα-Cα distances among two proteins. For two proteins, the larger the Z-score, the greater the similarity, which corresponds to the optimal set of residue equivalence obtained by permuting the equivalent structural patterns by Monte Carlo optimization. A Z-score < 2 is considered as a spurious similarity and should be ignored [37].

DALI supports three types of database searches (PDB, PDB25 and AlphaFold-Database (AF-DB)), as well as two types of structure comparisons (pairwise and all against all). Proteins in secondary structures are traditionally characterized with three states: (1) helix (H), strand (E) and Coil (C). The Dictionary of Secondary Structure of Proteins (DSSP) [38] offers a finer classification of the secondary structures by extending the three general states into eight states. DALI uses the secondary structure assignments by DSSP.

Beside DALI, other protein structures similarity searching and analysis tools such as PDBeFold [14] and FATCAT 2.0 [5] also provide the pairwise alignment and comparison for the similar protein structures. PDBefold can be used to search similar protein structures in the whole PDB and SCOP databases or any subset of SCOP. Whereas FATCAT searches for similar protein structures in the subsets of PDB, SCOP and ECOD [39] databases. Table 1 compares the characteristics of three above servers. Another recent tool GrAfSS [4] searches the PDB to identify known structural arrangements or 3D motifs in protein structures. However, GrAfSS cannot perform the pairwise structural similarity. The main reason to use DALI in this work is that it is developed purely for the protein structure similarity searching and analysis in the PDB database.

Generally, to find various kinds of patterns in different data types, multiple algorithms have been developed. Frequent Itemset Mining (FIM) [40] and Association Rule Mining (ARM) [41] are the two most popular pattern mining problems. FIM finds items (symbols) sets that have a support (frequency) that is equal to or greater than a minimum support threshold selected by the user. ARM's goal is similar to FIM. However, patterns in ARM are represented

Table 1: Comparison of three protein structure servers. Input types are represented as 1: PDB ID, 2: SCOP ID, 3: Protein structure coordinate file.

	DALI	**PDBeFold**	**FATCAT**
Input format	1, 3	1, 2, 3	1, 2, 3
Databases searched against	PDB, AF	PDB, SCOP	PDB, SCOP, ECOD
Multiple structural alignment	Yes	Yes	No
Dendrogram and Heatmap for multiple structures	Yes	No	No
Scoring functions used	Multiple	Multiple	Multiple

as rules rather than sets. ARM computes not only the support but also the confidence (an estimation of the conditional probability) of a rule.

However, both ARM and FIM do not work well on time-based data and do not take into account the sequential ordering of events such as the ordering of the nucleotides and AA in genome sequences. To address that issue with FIM and ARM, the task of SPM was proposed. SPM [33] analyze sequential data by discovering interesting (sub)sequences in a set of sequences. SPM can also analyze time series data. Various measures are used to evaluate the interestingness of a (sub)sequence. For example, the measure based on finding sub(sequences) occurrence frequency, (sub)sequences length, and the profit subsequences generate. As SPM can process data encoded as sequences of symbols or events, it has been used in many real life applications such as in bioinformatics [42, 43], proof sequence analysis [44], e-learning [45], text analysis [46, 47], energy reduction in smarthomes [48], malware detection [49], and webpage click-stream analysis [50].

The three steps of the proposed PSAC-PDB method are described in more details next.

2.1. Similar proteins Structures Identification in PDB

The SARS-CoV-2 protein structure with PDB ID (PID) 6VSB [35] (deposited to PDB on 10 February 2020) is used as the query structure. The main reason to select 6VSB as query structure is that it is one of the earliest S protein structures deposited in PDB. 6VSB contains three chains (A, B and C) and 1,288 AA. Using 6VSB Chain B as the query structure, DALI returned 397 structures via PDB90 search. PDB90 is a non-redundant subset of PDB structures. In a PDB90 subset, structures from PDB are found that are less than 90% identical as a sequence. After removing the same structures with different chains, the number of structures is reduced to 388. Note that PDBeFold found 1,156 similar structures against 6VSB Chain B and FATCAT found 91 similar structures against 6VSB Chain B in 90% non-redundant PDB dataset.

We also used BLAST [51], one of the most famous algorithm for biological sequence comparison, to see the first 100 structures that are the most similar to the query structure. Interestingly, all those 100 structures returned by BLAST belong to the S protein structures of SARS-CoV-2. Whereas with DALI, the similar protein structures against the query structure (6VSB with Chain B) varies from coronaviruses to various other enzymes and proteins. The similar structures obtained can be divided into three types (families):

1. S protein structures of SARS-CoV-2 (SSC2),
2. S protein structures of other viruses and organisms (SO), and
3. Protein (enzyme) structures for others (O).

In the similar protein structures obtained via the PDB90 search in DALI, approximately 13.65% (53 out of 388) belong to the first type (S protein structures of SARS-CoV-2), approximately 12.37% (48 out of 388) belong to the second type (S protein structures of other viruses and organisms). Remaining belong to the third type (structures of others). The AA sequences of all 388 structures are then obtained from the PDB. Some sequences have multiple AA sequences due to multiple chains. Thus, the downloaded sequences are refined to only include the sequences for the chain which is similar to the query structure. Table 2 provides statistics about the structures that belong to each of the three families.

Table 2: Structures distribution according to their families.

Structures	**Samples**	**AA**	**FAA**	**MinL, MaxL, ASL**
SSC2	53	20	L, S, T, V, G	127, 1380, 1074
SO	48	20	S, L, V, T, G	135, 1469, 847
O	287	22	L, G, S, V,A	69, 4646, 375
Total	388	22	L, S, G, V, T	69, 4646, 526

FAA: Frequent AA

For each structure type, the number of distinct AA, and the five most frequent AA are shown in the third and fourth columns of Table 2. On average, an AA sequence of SSC2, SO and O contain approximately 1,074, 847 and 375 AA. The five most frequent AA in SSC2 are Leucine (L) (8.30%), Serine (S) (8.05%), Threonine (T) (7.43%), Valine (V) (7.42%) and Glycine (G) (7.11%). In SO, the five most frequent AA are: S (8.52%), L (8.50%), Valine (V) (7.78%), T (7.39%) and G (6.76%). The five most frequent AA in O are: L (8.05%), G (7.62%), S (7.42%), V (6.73%) and Alanine (A) (6.48%). The third family, O, has 22 distinct AA because the amino acid B that can be either Asparagine (N) amino acid or Aspartic (D) amino acid is present twice in one structure (PDB ID: 2FMD) and the amino acid X that can be any of the 20 AA is present once in three structures (PDB IDs: 1C1F, 1UMZ, 4CG4), twice in one structure (PDB ID:3IPV) and six times in one structure (PDB ID: 3USU).

2.2. Protein Structures Analysis and Datasets Development

SPM is used on the AA sequences of three protein structures families to discover frequent AA and their patterns. To efficiently use SPM, AA sequences are first transformed into a suitable electronic format where the "*AAs to integers*" abstraction is used, where each AA is converted into a distinct positive integer. Some key concepts related to sequences and SPM are discussed next.

Let AA be the set of all distinct AA in sequences. An *amino acids set* AAS is a set of AA such that $AAS \subseteq AA$. The notation $|AAS|$ denotes the set cardinality. AAS has a length k (called k-AAS) if it contains k AA, i.e., $|AAS| = k$. For example, consider the set of $AA = \{A,\ V,\ C,\ T,\ N,\ K\}$. Then, the set $\{A,\ V,\ T,\ K\}$ is an AA set that contains four AA. On AA, a total order relation is defined for facilitating the discovery of patterns. This order is used as processing order for searching the patterns and is generally the lexicographical order. Now, an AA sequence is a list of AA sets $S = \langle AAS_1,\ AAS_2,\ ...,\ AAS_n \rangle$, such that $AAS_i \subseteq AAS$ $(1 \leq i \leq n)$. An *AA dataset* AAD is a list of AA sequences $AAD = \langle S_1, S_2, ..., S_n \rangle$, where each sequence has a unique identifier (ID). For instance, Table 3 shows an AAD that has five AA sequences.

Table 3: An AAD sample.

ID	**AA sequence**
1	⟨....VWTDLYGCLV.....⟩
2	⟨....LIQEVIFSTL.....⟩
3	⟨....DIQSMFYACN.....⟩
4	⟨....TCFGDNEIVQ.....⟩
5	⟨....LRPFERDISN.....⟩

To make an AAD more suitable for SPM, AA sequences are converted into sequences of integers. Thus, in the final AAD, each line represents the AA sequence for a protein structure and each AA is replaced by a unique positive integer. For example, the AA A and C are replaced by 1 and 3 respectively. Moreover, AA are separated from each other by a single space followed by a negative integer -1. A negative integer -2 is appended at the end of each line to indicate the end of the sequence. Table 4 shows the integer sequences for the AA sequences of Table 3.

Table 4: AA sequences conversion to integer sequences.

ID	**AA sequence**
1	22 -1 23 -1 20 -1 4 1 12 -1 25 -1 7 1 3 -1 12 -1 22 -1 -2
2	12 -1 9 -1 17 -1 5 -1 22 -1 9 1 6 -1 19 -1 20 -1 12 -1 -2
3	4 -1 9 -1 17 -1 19 -1 13 -1 6 1 25 -1 1 -1 3 -1 14 -1 -2
4	20 -1 3 -1 6 -1 7 -1 4 1 14 -1 5 1 9 -1 22 -1 17 -1 -2
5	12 -1 18 -1 ‘16 -1 6 -1 5 -1 18 1 4 -1 9 -1 19 -1 14 -1 -2

After the dataset development, the next step is to extract patterns from the AA sequences. An AA sequence $S_\alpha = \langle \alpha_1, \alpha_2, ..., \alpha_n \rangle$ is present or contained in another AA sequence $S_\beta = \langle \beta_1, \beta_1, ..., \beta_m \rangle$ iff there exist integers $1 \leq i_1 < i_2 < ... < i_n \leq m$, such that $\alpha_1 \subseteq \beta_{i1}, \alpha_2 \subseteq \beta_{i2}, ..., \alpha_n \subseteq \beta_{im}$ (denoted as $S_\alpha \sqsubseteq S_\beta$). If S_α is present in S_β, then S_α is a *subsequence* of S_β. In SPM, the *support* measure is used mostly to investigate the importance and interestingness of (sub)sequences. The *support* of S_α in AAD is the total number of sequences (S) that contain S_α, and is represented by $sup(S_\alpha)$:

$$sup(S_\alpha) = |\{S | S_\alpha \sqsubseteq S \wedge S \in AAD\}|$$

A sequence S is a *frequent sequence* (also called *sequential pattern*) iff $sup(S) \geq minsup$, where $minsup$ (minimum support) is a threshold determined by the user. An AA sequence can generally have up to $2^{20} - 1$ distinct subsequences. Thus, using the naive approach to compute the support of all possible subsequences is infeasible. However, various efficient algorithms are now present that can find all sequential patterns without exploring all the search space of all possible subsequences.

Two operations are used by SPM algorithms to traverse the search space of sequential patterns, that are: *s-extensions* and *i-extensions*. A sequential pattern $S_\alpha = \langle \alpha_1, \alpha_2, ..., \alpha_n \rangle$ is a *prefix* of another pattern $S_\beta = \langle \beta_1, \beta_1, ..., \beta_m \rangle$, if $n < m$, $\alpha_1 = \beta_1$, $\alpha_2 = \beta_2$, ..., $\alpha_{n-1} = \beta_{n-1}$, where α_n is equal to the first $|\alpha_n|$ items of β_n according to the $\prec$ order. Note that SPM algorithms follow a specific order $\prec$ to avoid considering the same potential patterns twice. The choice of $\prec$ does not affect the final result of SPM algorithms. For an item x, S_β is an *s-extension* of S_α if $S_\beta = \langle \alpha_1, \alpha_2, ..., \alpha_n, \{x\} \rangle$. Similarly, S_γ is an *i-extension* of S_α for an item x if $S_\gamma = \langle \alpha_1, \alpha_2, ..., \alpha_n \cup \{x\} \rangle$. SPM algorithms generally either employ a breadth-first search or depth-first search. Next, some SPM algorithms used in this work are briefly introduced.

For FIM, Apriori [52] is the first and most popular algorithm that can find frequent itemsets (e.g. sets of AA) in a database. Apriori starts by searching for items that occur frequently. These items are then extended to find larger itemsets that appear frequently enough. On the other hand, the TKS (Top-k Sequential) [53] and CM-SPAM [54] algorithms are some efficient and new algorithms. TKS finds the top-k sequential patterns in a database. The parameter k, set by the user, represents the number of sequential patterns that TKS needs to extract. TKS uses the basic candidate generation procedure of the SPAM algorithm and a vertical database representation. The vertical database representation allows to count the patterns support by avoiding costly database scans. Thus, vertical SPM algorithms are preferred in this work as the dataset contains long sequences of AA. TKS also uses many strategies to reduce the search space and utilize a PMAP (Precedence Map) data structure for avoiding the costly operations of bit vector intersection. CM-SPAM examines the whole search space to extract frequent sequential patterns in a dataset. CM-SPAM uses the CMAP (Co-occurrence MAP) data structure to store information related to co-occurrences of items. CM-SPAM can efficiently discover sequential patterns as it employs a vertical database representation and uses a powerful search space pruning mechanism.

AA sequences of protein structures and sequential frequent patterns discovered in them by using TKS and CM-SPAM are used separately to classify protein structures to different types. Moreover, DALI is used to not only find similar protein structures but for the structural alignment of similar protein structures too. In DALI, the main measure used to assess similarity between protein structures is the Z-score. DALI provides two blocks in structural alignment: one for the aligned AA (AAA) sequences and one for the aligned SSE (ASSE). Thus, AA sequences (downloaded from PDB), FAA patterns (discovered by using SPM algorithms) and AAA and ASSE sequences (discovered through DALI) of protein structures are stored in respective datasets. Thus, we have four datasets for protein structures: (1) AA sequences, (2) FAA, (3) AAA and (4) ASSE.

2.3. Classification

The third step is to classify protein structures into three types using the four datasets of AA, FAA, AAA and ASSE sequences.

AA sequences of proteins are generally long (see Table 2). A close inspection of the AA sequences dataset revealed that almost all sequences contain the same AA tens or even hundreds of times, sometimes repeated consecutively. We believe that this repetition of AA in protein sequences can be replaced with frequent sequential AA for better classification performance. In the results section, we found that the resulting frequent patterns indeed provided better classification performance. More precisely, PSAC-PDB utilizes the frequent sequential patterns in AA sequences obtained by using two SPM algorithms for the classification of protein structures.

Two standard methods are used for classification, which are binary classification and multi-class (MC) classification. Binary classification is done on four datasets to train a model to classify three structure types separately. For a selected structure type, binary classification (Definition 1) assigns "class name" to each sequence from four datasets corresponding to that type and label all other sequences as "Others".

Definition 1. *Let C be the set of three protein structures classes (types). For a selected protein structures class $c \in C$, a sequence Y, in four datasets, is labeled with respect to c as:*

$$Y_c = \begin{cases} c, & \text{if } y = c, \\ Others, & \text{otherwise} \end{cases} \tag{1}$$

According to Equation 1, class labels that belong to c will be labeled as c, while others are labeled as $Others$ to then train a binary classifier. For example, for the first type of protein structures, Equation 1 will assign "$SSC2$" to AA, FAA, AAA and ASSE sequences that belong to that type and "$Others$" to AA, FAA, AAA and ASSE sequences that belong to the other two types.

On the other hand, in multi-class classification, the three types of protein structures in four datasets are labeled with their respective class name. This means that in the four datasets, the sequences that belong to the first, second and third types are labelled as SSC2, SO, and O respectively.

Six metrics are used to evaluate and compare the performance of classifiers, which are: (1) accuracy, (2) false positive rate (FPR), (3) recall, (4) precision, (5) F1 score and (6) Matthews correlation coefficient (MCC). In this work, the accuracy (ACC) is defined as the percentage of correctly classified protein structures types divided by the total number of protein structure types. Thus, ACC is defined as:

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

where TP stands for true positives (number of protein structures correctly identified as belonging to a given protein structure type), FP is the false positives (number of protein structures incorrectly identified as belonging to a given protein structure type), FN is the false negatives (number of protein structures incorrectly identified as not belonging to a given protein structure type) and TN is the true negatives (number of protein structures correctly identified as not belonging to a given protein structure type).

The other five measures, FPR, precision, recall, f-measure and MCC are calculated as follows:

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

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

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

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

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

Eleven ML algorithms are used for classification, which are: (1) NB (Naive Bayes), (2) SVM (Support Vector Machine), (3) kNN (k-Nearest Neighbors), (4) KStar (K*), (5) J48 (Decision Tree), (6) Random Forest (RF), (7) Logistic Regression (LR), (8) CNN (Convolutional Neural Network), (9) ZeroR, (10) MNBT (Multinomial Naive Bayes Text) and (11) Stochastic Gradient Descent Text (SGDT) [55, 56].

The first eight (NB, SVM, kNN, K*, J48, RF, LR and CNN) are integer-based classifiers and the last three (ZeroR, MNBT, SGDT) are text (string)-based classifiers. Four different tokenization strategies are used in string-based classifiers, that are (1) Word Tokenizer, (2) NGram Tokenizer, (3) Character NGram Tokenizer and (4) Alphabetic Tokenizer. The effect of those four tokenizers on the performance of classifiers is discussed in the next section. Standard 10-fold cross validation is used to evaluate the performance of the classifiers. Users and interested readers can use the PSAC-PDB as follows: First find the similar structure against a particular structure of interest (called the query structure) by using DALI. Identified similar structures can then be analyzed in DALI for multiple structural alignment, structures superimposition, phylogenetic analysis, etc. The AA of protein structures can then be analyzed with SPM algorithms. The AA, AAA, ASSE and FAA can then be used for the classification of identified protein structures. DALI is available online[2] and SPM algorithms implementations can be downloaded using SPMF software.

3. Results

The experiments were done on a workstation with a fifth-generation Core i7 processor and 32 GB of RAM. The SPMF data mining library [57], developed in Java, was used to analyze and discover patterns in the dataset of AA sequences. SPMF is an open-source cross-platform framework, specialized in pattern mining tasks, which offers implementations of more than 230 data mining algorithms. The open-source WEKA software [58], developed in Java, was used to train and test the classifiers on four datasets. WEKA was selected because it can run on various platforms and offers not only classifiers for machine learning but also tools for data preparation and meta learners. Moreover, it also provides a graphical interface, along with its command line interface, that is easy to use. Results obtained by applying the SPM algorithms on the AA sequences dataset are discussed next, followed by DALI and classification results.

3.1. Frequent Patterns

First, the Apriori algorithm is applied on the AA dataset for protein structures of three types to find frequently occurring AA (Table 5). The top five AA in the whole dataset (named All that contains AA sequences of all protein structures) are: L, S, G, V and T. Whereas in the AA sequences dataset for protein structures of three types SSC2(SO) and O, the top five AA are: L, S, T, V, G (S, L, V, T, G) and L, G, S, V, A respectively. The frequent sets of AA discovered by Apriori are unordered. Moreover, Apriori does not ensure that AA from an AA set appear contiguously in a sequence. Thus frequent patterns of larger length discovered by Apriori are uninteresting and do not provide any useful information. Note that Apriori cannot discover sequential patterns as it ignores the sequential relationship among AA.

SPM algorithm such as TKS and CM-SPAM overcome the drawbacks of Apriori as they can discover more meaningful patterns and information. The TKS algorithm is applied on the AA sequences dataset to find the top-k sequential patterns of AA. Unlike TKS, the CM-SPAM algorithm requires setting the *minsup* threshold. Some AA frequent patterns discovered by the TKS and CM-SPAM algorithms, in the whole dataset and in the dataset for three types, with varying length are shown in Table 6. Table 6 provides some useful information related to frequent occurrences of AA patterns in the protein structures. Note that patterns discovered by the CM-SPAM algorithm are almost the same as those obtained by the TKS algorithm.

Overall, we found that the pattern mining process was quite fast. It is observed that by decreasing *minsup* in Apriori and CM-SPAM, and increasing the

[2] ekhidna2.biocenter.helsinki.fi/dali

Table 5: Extracted frequent AA.

AA	Frequency(All)	Frequency(SSC2)	Frequency(SO)	Frequency(O)
A	13,093	3,568	2,546	6,979
C	3,559	1,426	9,96	1,137
D	11,417	2,724	2,026	6,667
E	10,113	2,189	1,565	6,359
F	10,547	3,452	2,300	4,795
G	15,005	4,051	2,747	8,207
H	4,477	1,036	7,84	2,657
I	11,483	3,185	2,374	5,971
K	10,286	2,631	1,638	6,017
L	16,863	4,733	3,453	8,677
M	3,096	5,38	5,45	2,013
N	12,736	3,904	2,706	6,126
P	9,397	2,776	1,777	4,844
Q	8,983	2,824	1,883	4,276
R	8,002	1,909	1,450	4,643
S	16,044	4,585	3,462	7,997
T	14,024	4,232	3,003	6,789
V	14,647	4,229	3,161	7,257
W	2,950	5,24	3,67	2,059
Y	8,529	2,440	1,886	4,203

Table 6: Extracted frequent sequential AA patterns by using the TKS and CM-SPAM.

	All	SSC2	SO	O
TKS	KLS	AAL	YYR	CAG
	ELVL	GLAE	VUDD	VUDD
	DAGFI	LLKLL	TLAPD	AGLAD
	GLAEEL	CDEIPI	SLTDDV	GRGLVP
	SLLESLL	FAQQVKN	FVREFNK	HHHHHHS
	ANQFNSAI	AAAYYYV	KTPQMYTLK	QTVAVEFD
	TLADAGFIK	HADQLWPTP	PDPLKNTKR	SGLVNRGSN
	VLPPLLTDEM	IADTTDAVRD	NNYPAIPTND	WEDIDIEFLG
	IVNNTVYDPLQ	KAHFPRDGFA	MAYRFNGIGEL	FLGKDTTKFQV
	CGPKKSTNLVKN	LKPFERDISTD	LRPFERDISNVA	DEIDIEFLGKED
CM-SPAM	LLS	AAE	YYR	CAG
	ELLL	CDIP	VYDP	DEFD
	CPFGE	DAGFI	TVYDP	ADGLA
	QPTESI	ECDIPI	SITTEV	GLVPRG
	RDIADTT	ECDIPI	SITTEV	GLVPRG
	NCTEVPVA	GAAAYYVG	QMYKTPTLK	QTVAVEFF
	TLADAGFIK	HADQLTPTW	QMYKTPTLK	QTVAVEFF
	AENSVAYSNN	AADTTAAVRD	NNTIAIPTNF	WDEIDIEFLG
	FTISVTTEILP	KAHFPREGVF	MAYRFNGIGVT	FLGKDTTKVQF
	CGPKKSTNLVKN	LKPFERDISTE	LRPFERDISNVP	DEIDIEFLGKDT

k parameter of TKS, more frequent patterns can be discovered, while the runtime and the memory usage increases. During execution, the three algorithms worked efficiently on the dataset.

3.2. DALI Results

The results of the pairwise sequence alignment by using DALI on 12 protein structures against the query structure is presented. DALI aligned more than 970 AA in 12 structures and the first 300 AA alignment is shown in the upper block of Figure 2. The most frequent AA in each column (structure) are colored. The uppercase letters represent those positions that are structurally equivalent with the query structure. The second part (lower block) in Figure 2 shows the secondary structure states. The most frequent SSE are: Coil or turn (L), followed by α-helix (H) and β-sheet (E). Note that two structures (6NB7 and 6CS2) belong to SARS-CoV, 6JX7 to Feline infectious peritonitis (FIP) virus, 5I08 to Human Coronavirus-HKU1, 6M5Y to sugar binding protein and 50CQ to the hydrolase enzyme.

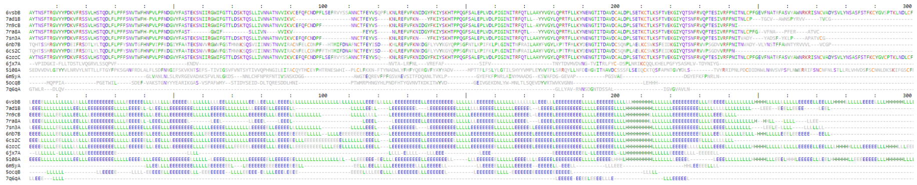

Figure 2: AAA and ASSE in 12 structures obtained by using DALI. 6VSB is used as the query structure.

DALI also reports the root-mean square deviation (RMSD) of aligned Cα-atoms, LALI (number of aligned Cα-atoms), NRES (number of AA residues in the target structure) and IDEN (% identity of AAA) for similar protein structures. Table 7 lists the DALI's outputs for 12 protein structures against the query structure. High Z-score, LALI, NRES and low RMSD means that the structure is most similar (high IDEN) to a query structure (6VSB here). Note that DALI does not generate alignments with low RMSD as it maximizes the Z-score (a geometrical similarity score), defined in terms of similarities of intramolecular distances. An alignment is considered "better" if it has both smaller RMSD and larger LALI. If both RMSD and LALI are smaller or larger, it is not possible to establish an order between the alignments. The AAA and ASSE of protein structures are also used for the classification.

Figure 3 compares the first similar 100 protein structures on the basis of their Z-score (blue colored line), RMSD (orange colored line), LALI (black colored Bar), NRES (gray colored bar) and IDEN (red colored line). Generally high Z-score and LALI mean that the protein structures are most similar and the opposite is true for RMSD. We can see that as we move from right to left (from most similar to dissimilar), RMSD value increases whereas LALI and Z-score decreases. In some cases (left side), protein structures are more similar even for low Z-score and LALI. This shows that only one measure, particularly Z-score or RMSD cannot determine the (dis)similarity of protein structures.

Table 7: DALI results for 12 structures against the query structure.

Structures PID	Z-score	RMSD	LALI	NRES	IDEN
7AD1B	48.8	3.6	792	935	97
7Q6QA	48.2	1.6	965	1013	99
7RA8A	48.1	2.7	762	917	97
7N9CB	48.1	1.7	778	812	99
7SN3A	47.8	2.9	803	925	97
6NB7B	40.7	5.0	822	1032	77
6CS2C	35.7	2.0	797	893	78
6ZOZC	32.9	3.3	964	1070	99
6JX7A	28.0	8.7	604	1245	28
5I08A	27.4	3.9	801	958	31
6M5YA	8.4	8.1	132	270	8
5OCQB	5.7	3.8	136	279	6

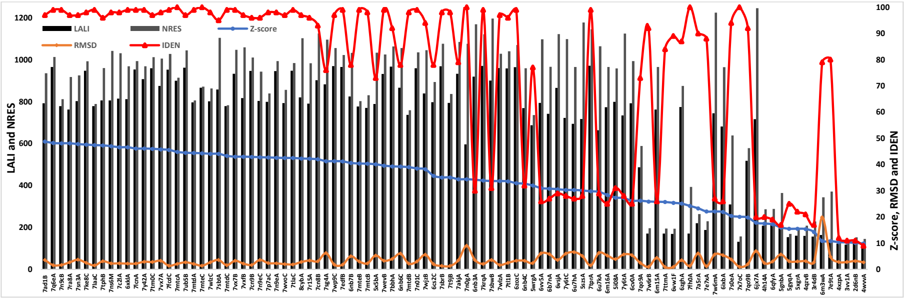

Figure 3: Comparison of first 100 similar protein structures against the query structure.

3.3. *Classification Results*

Three text-based classifiers (MNBT, SGDT and ZeroR) are used for the binary and multi-class (MC) classification in three datasets (AA, AAA and ASSE). Whereas for the dataset of FAA patterns identified by TKS and CM-SPAM, both integer and text-based classifiers are used.

3.3.1. *AA, AAA and ASSE*

MNBT and SGDT are used with four tokenization strategies: (1) Word-Tokenizer (WT), (2) NGram-Tokenizer (NGT),(3) CharacterNGramTokenizer (CNGT) and (4) AlphabeticTokenizer (AT). The first one is a simple technique to tokenize the strings. The second tokenizer splits a string into an n-gram with user specified minimum and maximum grams. Whereas, the third tokenizer splits a string into all character n-grams it contains on the basis of user specified maximum and minimum values for n. In both NGT and CNGT, the maximum and minimum grams were set to 3 and 1 respectively. The fourth tokenizer forms tokens from contiguous alphabetic sequences only. Table 8 provides the results for the classifier metrics with the following format: $AA\left(\frac{AAA}{ASSE}\right)$. For example, the first entry $86.34\left(\frac{86.30}{86.30}\right)$ indicates that for the SSC2 type, MNBT achieved ACC of 86.34% on the AA dataset and 86.30% ACC on AAA and

ASSE datasets respectively. Three strategies WT, NGT and AT generated the same results for both classifiers. Whereas the CNGT strategy performed better than WT, NGT and AT on both classifiers. The results for the ZeroR on AA, AAA and ASSE for various parameters were the same as NMBT's results with WT, NGT and AT strategies. On the three datasets (for AA, AAA and ASSE), SGDT with CNGT strategy performed better, overall, than MNBT with the same strategy for binary classification. For multi-class classification, results are provided for MNBT as SGDT cannot be used for this type of classification.

Table 8: Classifiers performance on three datasets (AA, AAA and ASSE) with different tokenization strategies.

Type	**P**	**MNBT**		**SGDT**	
		WT*	**CNGT**	**WT****	**CNGT**
SSC2	ACC	$86.34(\frac{86.30}{86.30})$	$94.84(\frac{96.12}{93.28})$	$87.62(\frac{86.30}{86.30})$	$\mathbf{96.39(\frac{96.89}{91.98})}$
	FPR	$0.863(\frac{0.863}{0.863})$.	$0.119(\frac{0.022}{0.185})$	$0.782(\frac{0.863}{0.863})$	$\mathbf{0.0117(\frac{0.084}{0.020})}$
	P	$?(\frac{?}{?})$	$0.952(\frac{0.968}{0.935})$	$0.892(\frac{?}{?})$	$\mathbf{0.964(\frac{0.970}{0.925})}$
	R	$0.863(\frac{0.863}{0.863})$	$0.948(\frac{0.961}{0.933})$	$0.876(\frac{0.863}{0.863})$	$\mathbf{0.964(\frac{0.969}{0.920})}$
	F1	$?(\frac{?}{?})$	$0.950(\frac{0.963}{0.934})$	$0.829(\frac{?}{?})$	$\mathbf{0.964(\frac{0.969}{0.922})}$
	MCC	$?(\frac{?}{?})$	$0.793(\frac{0.859}{0.725})$	$0.287(\frac{?}{?})$	$\mathbf{0.847(\frac{0.871}{0.681})}$
SO	ACC	$87.62(\frac{87.59}{87.59})$	$92.78(\frac{91.98}{93.28})$	$87.62(\frac{87.59}{86.30})$	$\mathbf{93(\frac{92.50}{88.63})}$
	FPR	$0.876(\frac{0.876}{0.876})$.	$0.386(\frac{0.280}{0.124})$	$0.876(\frac{0.876}{0.876})$	$\mathbf{0.403(\frac{0.386}{0.392})}$
	P	$?(\frac{?}{?})$	$0.923(\frac{0.921}{?})$	$?(\frac{?}{?})$	$\mathbf{0.926(\frac{0.919}{0.888})}$
	R	$0.876(\frac{0.876}{0.876})$	$0.928(\frac{0.920}{0.124})$	$0.876(\frac{0.876}{0.876})$	$\mathbf{0.930(\frac{0.926}{0.886})}$
	F1	$?(\frac{?}{?})$	$0.922(\frac{0.920}{?})$	$?(\frac{?}{?})$	$\mathbf{0.924(\frac{0.920}{0.887})}$
	MCC	$?(\frac{?}{?})$	$0.631(\frac{0.635}{?})$	$?(\frac{?}{?})$	$\mathbf{0.640(\frac{0.619}{0.486})}$
O	ACC	$73.96(\frac{73.90}{73.90})$	$90.20(\frac{93.28}{86.21})$	$75.25(\frac{73.90}{73.90})$	$\mathbf{95.3(\frac{95.09}{88.11})}$
	FPR	$0.740(\frac{0.739}{0.739})$.	$0.278(\frac{0.177}{0.245})$	$0.703(\frac{0.739}{0.739})$	$\mathbf{0.119(\frac{0.113}{0.881})}$
	P	$?(\frac{?}{?})$	$0.914(\frac{0.936}{0.865})$	$0.815(\frac{?}{?})$	$\mathbf{0.955(\frac{0.951}{0.885})}$
	R	$0.740(\frac{0.739}{0.739})$	$0.902(\frac{0.933}{0.868})$	$0.753(\frac{0.739}{0.739})$	$\mathbf{0.954(\frac{0.951}{0.881})}$
	F1	$?(\frac{?}{?})$	$0.894(\frac{0.930}{0.866})$	$0.658(\frac{?}{?})$	$\mathbf{0.952(\frac{0.950}{0.883})}$
	MCC	$?(\frac{?}{?})$	$0.742(\frac{0.832}{0.648})$	$0.193(\frac{?}{?})$	$\mathbf{0.878(\frac{0.871}{0.701})}$
MC	ACC	$73.96(\frac{73.90}{73.90})$	$92.26(\frac{94.31}{21.18})$	$-(\frac{-}{-})$	$-(\frac{-}{-})$
	FPR	$0.740(\frac{0.739}{0.739})$.	$0.157(\frac{0.099}{0.124})$	$-(\frac{-}{-})$	$-(\frac{-}{-})$
	P	$?(\frac{?}{?})$	$0.921(\frac{0.942}{?})$	$-(\frac{-}{-})$	$-(\frac{-}{-})$
	R	$0.740(\frac{0.739}{?})$	$0.923(\frac{0.943}{?})$	$-(\frac{-}{-})$	$-(\frac{-}{-})$
	F1	$?(\frac{?}{?})$	$0.916(\frac{0.939}{0.212})$	$-(\frac{-}{-})$	$-(\frac{-}{-})$
	MCC	$?(\frac{?}{?})$	$0.829(\frac{0.880}{?})$	$-(\frac{-}{-})$	$-(\frac{-}{-})$

*: MNBT performed similarly on WT, NGT and AT strategies.

**: SGDT performed similarly on WT, NGT and AT strategies.

Using AA sequences of proteins structures for binary classification, MNBT (SGDT) achieved 94.84%(96.39%), 92.78%(93%) and 90.20%(95.3%) for three types respectively with CNGT strategy. For Multi-class classification, MNBT with CNGT strategy yield 92.26% accuracy on the AA dataset. Except for the SO type, MNBT and SGDT accuracy is improved further on AAA dataset, which shows that AAA provide more reliable information for classification. On the other hand, using ASSE sequences for classification generated poor results (except for MNBT with CNGT strategy on SO) compared to AA and AAA sequences, particularly for multi-class classification. The main reason for this is that ASSE sequences only contain three elements. SGDT with different strategies on three datasets was slow (took minutes for training and testing) compared to MNBT (took seconds for training and testing).

Paired t-test (corrected) in WEKA is used to check which of three text-based classifiers are significantly better than others. ZeroR is selected as the baseline. Both MNBT and SGDT with CNGT strategy performed significantly better than ZeroR. For MNBT and SGDT, the later performed better than the former on the three datasets for binary classification. However, the test results confirmed that the difference in the performance of MNBT and SGDT on three datasets is not that significant.

3.3.2. FAA

TKS and CM-SPAM were used to discover frequent 100 and 200 sequential patterns in the AA dataset. Thus we have four sub-datasets for FAA: TKS100, TKS200, CM-SPAM100 and CM-SPAM200. These sub-datasets are further processed to make sure that the patterns contain (1) 5 to 10 (5-10) and (2) 10 to 15 (10-15) frequent AA in each pattern. The reason to consider two different numbers of discovered patterns (100 and 200) and two different number for the patterns length (5-10 and 10-15) is to investigate their effect on the classifiers performance.

Obtained accuracy of the binary and multi-class classification for patterns discovered with TKS and CM-SPAM are provided in Table 9 and Table 10 respectively. The results for classifiers metrics are shown with the following format: $\frac{100.5-10(100.10-15)}{200.5-10(200.10-15)}$. For example, consider the NB ACC of $\frac{68.66(69.33)}{62.16(64.33)}$ in Table 9. It indicates that an ACC of 68.66% is obtained on 100 patterns where each pattern contains 5-10 frequent AA discovered by TKS, 69.33% ACC is obtained using 100 patterns and 10-15 frequent AA in each pattern identified by TKS, 62.16% ACC is obtained using 200 patterns and 5-10 frequent AA in each pattern from TKS and 64.33% ACC using 200 patterns and 10-15 frequent AA in each pattern found by TKS, respectively. This format for metrics is used to reduce the total number of Tables.

All eight integer-based classifiers performed better on patterns discovered by CM-SPAM as compared to TKS. Tree based classifiers (RF and J48) performed better than NB, SVM, kNN, CNN and LR on patterns discovered by TKS. K* performed better than J48 on TKS patterns while NB performed less well than others. To use the text-based classifiers on FAA patterns, each integer in the corpus is replaced with its respective AA letter. The main reason to use text-based classifiers is to check whether they perform better than integer-based classifiers. Both MNBT and SGDT with CNGT strategy performed better than integer-based classifiers on FAA patterns discovered by using TKS and SGDT performed better than MNBT. On FAA patterns discovered by using CM-SPAM, K* and RF performance is almost similar with negligible difference. RF performance is better than MNBT and almost similar to the SGDT with negligible difference. Tree-based classifiers performed better because all the patterns in the datasets are used in the classification process where each pattern

Table 9: Classifiers accuracy results on patterns extracted by TKS.

Classifier	**SSC2**	SO	**O**	**MC**
NB	$\frac{68.66(69.33)}{62.16(64.33)}$	$\frac{64.66(70)}{63.33(62.66)}$	$\frac{53.36(84.66)}{56(71)}$	$\frac{47.66(60)}{38.5(40.83)}$
SVM	$\frac{72(67)}{66.66(66.66)}$	$\frac{66.66(66.66)}{66.66(66.66)}$	$\frac{66.66(70)}{66.66(66.66)}$	$\frac{41(50.33)}{37.66(39.16)}$
kNN	$\frac{72.66(93)}{76.33(68.66)}$	$\frac{68.33(95)}{70.5(68.16)}$	$\frac{70.66(94.33)}{70.16(77.83)}$	$\frac{60.66(91)}{59.5(58.16)}$
K*	$\frac{92.33(98.33)}{91.66(72.16)}$	$\frac{88.66(99.33)}{90.33(78.33)}$	$\frac{90.33(99)}{90.5(89.5)}$	$\frac{86.66(98.66)}{87.33(69.33)}$
J48	$\frac{87(96)}{88.5(72.66)}$	$\frac{88.33(96)}{88.5(72.66)}$	$\frac{87(97.33)}{84.33(89.16)}$	$\frac{82.33(95.66)}{80.16(64.66)}$
RF	$\frac{93.33(98)}{92.67(72.16)}$	$\frac{92(98.6)}{91(73.33)}$	$\frac{91.66(99.3)}{93(95.33)}$	$\frac{90(98)}{89.83(69)}$
CNN	$\frac{71.66(67.33)}{65.5(66.83)}$	$\frac{65.33(63)}{66.5(65.5)}$	$\frac{62(75.33)}{67.66(67.66)}$	$\frac{43.33(49)}{35.33(40)}$
LR	$\frac{71(67)}{65.16(66.5)}$	$\frac{64.66(64)}{66.66(65.5)}$	$\frac{63(76.66)}{67.33(67.5)}$	$\frac{41.33(47)}{35.33(39.5)}$
MNBT	$\frac{94.66(99.66)}{89.33(88.83)}$	$\frac{92.33(99.66)}{91.66(89.33)}$	$\frac{94.66(99.66)}{96.83(98.5)}$	$\frac{92.33(99.33)}{90.5(88.83)}$
SGDT	$\frac{98.66(99.66)}{96.33(85.83)}$	$\frac{96.66(99.66)}{94.5(86.83)}$	$\frac{96.66(99.66)}{94.5(99)}$	$\frac{-(-)}{-(-)}$

only contain FAA, which are considered as features. CNN classifier was slowest among all integer-based classifiers.

Table 10: Classifiers accuracy results on patterns extracted with CM-SPAM

Classifier	**SSC2**	SO	**O**	**MC**
NB	$\frac{98.33(97.33)}{90.33(71.5)}$	$\frac{79(90.66)}{88.66(88.66)}$	$\frac{75.33(82.33)}{75.66(69.83)}$	$\frac{79(85.66)}{80.33(63.16)}$
SVM	$\frac{97(93.33)}{90.83(73.83)}$	$\frac{73.33(83.33)}{91.83(92.5)}$	$\frac{72.66(66.33)}{77.83(70.16)}$	$\frac{73.66(75)}{83.83(67.83)}$
kNN	$\frac{96.66(93.66)}{82.66(84.5)}$	$\frac{87.33(95.66)}{92(95.83)}$	$\frac{85.66(92)}{80.16(83.83)}$	$\frac{86.33(91.66)}{71.86(81.5)}$
K*	$\frac{100(99.33)}{86.83(88.33)}$	$\frac{96.33(99)}{98(99.16)}$	$\frac{95.66(97.66)}{87(88.5)}$	$\frac{89.66(98)}{86.16(87.16)}$
J48	$\frac{99.33(99.33)}{92(90.16)}$	$\frac{94.66(98.66)}{98(98)}$	$\frac{94.66(97.66)}{93.16(89.16)}$	$\frac{93.66(98)}{93.16(89.16)}$
RF	$\frac{99.33(99.66)}{88(88.66)}$	$\frac{96(99.66)}{98.66(99.16)}$	$\frac{96.33(98)}{88.16(88.5)}$	$\frac{96(99)}{88(88.33)}$
CNN	$\frac{93.66(92)}{90.5(70.83)}$	$\frac{73.33(82.33)}{90.83(93.33)}$	$\frac{80(61.66)}{76.5(68.16)}$	$\frac{73.66(75.66)}{82.83(66)}$
LR	$\frac{96.66(97)}{89.66(71.66)}$	$\frac{79(62)}{91.33(93)}$	$\frac{73.33(62)}{76.16(68.5)}$	$\frac{79(78.66)}{83.16(66)}$
MNBT	$\frac{97(93.66)}{86.16(87.33)}$	$\frac{95.33(99)}{93.5(89.83)}$	$\frac{93.66(93)}{86.83(89.83)}$	$\frac{93(93.33)}{83.5(86.83)}$
SGDT	$\frac{100(99)}{88(88.33)}$	$\frac{98(99.66)}{98.83(99.66)}$	$\frac{99.33(99)}{87.66(89.16)}$	$\frac{-(-)}{-(-)}$

For TKS patterns, most of the classifiers performed better, overall, on 100 patterns as compared to 200 patterns. Moreover, classifiers performed better when patterns contained more FAA (10-15) compared to patterns that contain less FAA (5-10). For CM-SPAM, most of the classifiers performed better on 100 sequences as compared to 200 for SSC2, O types and multi-class classification. For the SO type, the opposite, classifiers performed better on 200 FAA patterns than 100 FAA patterns, is true. The same trend was observed for the patterns length. All classifiers performed better on 10-15 FAA for SSC2, O types and multi-class classification. For SO type, classifiers performed better on 5-10 FAA. Overall, RF performed better than other integer-based classifiers with both TKS and CM-SPAM patterns. Compared to text-based classifiers, RF performed better (similar) than MNBT(SGDT) with CM-SPAM patterns. The complete results of RF on TKS and CM-SPAM patterns are provided in Table 11 and

12 respectively. Note that RF generated high values of MCC meaning that it was able to correctly predict in most of the four categories of confusion matrix (TP, FN, TN and FP) even when the O type sequences are more (287) than SSC2 (53) and SO (48). These results indicate that most of the integer-based classifiers and text-based classifiers generated better results for classification of protein structures by using FAA as compared to the text-based classifiers results obtained on AA and AAA datasets

Table 11: Results for RF on patterns extracted using TKS.

P	**SSC2**	SO	**O**	**MC**
ACC	$\frac{93.33(98)}{92.67(72.16)}$	$\frac{92(98.6)}{91(73.33)}$	$\frac{91.66(99.3)}{93(95.33)}$	$\frac{90(98)}{89.83(69)}$
FPR	$\frac{0.123(0.030)}{0.124(0.437)}$	$\frac{0.120(0.012)}{0.138(0.338)}$	$\frac{0.127(0.013)}{0.125(0.071)}$	$\frac{0.050(0.010)}{0.051(0.155)}$
P	$\frac{0.937(0.980)}{0.928(0.708)}$	$\frac{0.920(0.987)}{0.910(0.767)}$	$\frac{0.916(0.993)}{0.932(0.953)}$	$\frac{0.901(0.980)}{0.899(0.687)}$
R	$\frac{0.933(0.980)}{0.927(0.722)}$	$\frac{0.920(0.987)}{0.910(0.773)}$	$\frac{0.917(0.993)}{0.930(0.953)}$	$\frac{0.900(0.980)}{0.898(0.690)}$
F1	$\frac{0.932(0.980)}{0.925(0.703)}$	$\frac{0.919(0.987)}{0.909(0.765)}$	$\frac{0.916(0.993)}{0.928(0.953)}$	$\frac{0.900(0.980)}{0.898(0.688)}$
MCC	$\frac{0.850(0.955)}{0.834(0.703)}$	$\frac{0.818(0.970)}{0.795(0.467)}$	$\frac{0.810(0.985)}{0.842(0.894)}$	$\frac{0.850(0.970)}{0.848(0.534)}$

Table 12: RF results on patterns extracted with CM-SPAM.

P	**SSC2**	SO	**O**	**MC**
ACC	$\frac{99.33(99.66)}{88(88.66)}$	$\frac{96(99.66)}{98.66(99.16)}$	$\frac{96.33(98)}{88.16(88.5)}$	$\frac{96(99)}{88(88.33)}$
FPR	$\frac{0.003(0.002)}{0.148(0.139)}$	$\frac{0.060(0.002)}{0.024(0.017)}$	$\frac{0.038(0.030)}{0.152(0.145)}$	$\frac{0.020(0.005)}{0.060(0.058)}$
P	$\frac{0.993(0.997)}{0.880(0.887)}$	$\frac{0.960(0.997)}{0.987(0.992)}$	$\frac{0.964(0.980)}{0.881(0.885)}$	$\frac{0.960(0.990)}{0.882(0.884)}$
R	$\frac{0.993(0.997)}{0.880(0.887)}$	$\frac{0.960(0.997)}{0.987(0.992)}$	$\frac{0.963(0.980)}{0.882(0.885)}$	$\frac{0.960(0.990)}{0.880(0.883)}$
F1	$\frac{0.993(0.997}{0.880(0.887)}$	$\frac{0.960(0.997)}{0.987(0.992)}$	$\frac{0.963(0.980)}{0.881(0.885)}$	$\frac{0.960(0.990)}{0.881(0.884)}$
MCC	$\frac{0.985(0.993)}{0.731(0.746)}$	$\frac{0.910(0.993)}{0.970(0.981)}$	$\frac{0.918(0.955)}{0.733(0.741)}$	$\frac{0.940(0.985)}{0.821(0.825)}$

We also performed paired t-test (corrected) in WEKA to check whether RF is significantly better than other seven integer-based classifiers or not. The comparison results for the ACC of classifiers is provided in Table 13. The entries before bracket are for the TKS and in the brackets are for CM-SPAM. Bold entries for classifiers are those that performed significantly lower than RF. Whereas the Underline entries for classifiers show that they performed significantly better than RF. K* performed almost similarly to RF on patterns discovered with TKS and CM-SPAM. This means that the difference in the performance of K* and RF is not that significant in most cases. Whereas J48 performed significantly better than RF on the CM-SPAM 200 patterns with each pattern containing 5-10 for multi-class classification, SSC2 and O binary classification.

In summary, overall results show that frequent sequential AA patterns can be used more efficiently for the classification and detection of protein structures in place of providing the whole AA sequences, AAA and ASSE. From Table 2 we can see that protein structures contain hundreds of AA on average in each

Table 13: Paired t-test results for classifiers.

Dataset		RF	NB	SVM	kNN	K*	J48	LR	CNN
MC	1	88.97(96.43)	**47.77**(**78.20**)	**40.40**(**74.63**)	**60.67**(**86.63**)	86.23(**96.33**)	**80.27**(**93,27**)	**41.23**(**78.97**)	**42.07**(**73.90**)
	2	97.60(99.10)	**59.73**(**85.93**)	**50.57**(**75.90**)	**90.50**(**91.63**)	98.23(97.93)	**94.97**(**98.40**)	**46.83**(**80.17**)	**46.83**(**75.07**)
	3	89.13(87.55)	**38.35**(**80.23**)	**37.32**(**84.25**)	**58.12**(**77.47**)	**86.48**(**85.73**)	**81.30**(92.60)	**36.33**(**83.45**)	**36.35**(**83.05**)
	4	71.52(88.18)	**41.78**(**63.37**)	**40.93**(**68.17**)	**58.12**(**81.63**	69.55(87.42)	**62.83**(89.30)	**40.27**(**66.37**)	**40.40**(**67.13**)
SSC2	1	94.27(99.40)	**67.67**(98.30)	**71.90**(**96.73**)	**78.40**(**96.63**)	93.43(99.87)	**89.43**(99.33)	**71.10**(97.77)	**71.37**(**93.73**)
	2	98.50(99.43)	**69.80**(**96.97**)	**68.13**(**93.13**)	**93.60** (**94.17**)	98.87(99.03)	**95.73**(99.33)	**67.40**(**96.67**)	**67.60**(**91.97**)
	3	92.08(88.12)	**63.67**(90.35)	**66.67**(90.97)	**75.45**(**83.38**)	91.45(87.30)	**87.18**(91.47)	**65.15**(90.13)	**65.05**(90.33)
	4	70.95(88.57)	**64.77**(**72.28**)	**66.67**(**74.32**)	68.15(**84.22**)	72.00(88.15)	68.33(90.02)	**66.13**(**72.55**)	**65.90**(**72.23**)
SO	1	92.07(96.27)	**64.33**(**77.30**)	**66.10**(**78.90**)	**70.03**(**89.17**)	89.63(96.37)	**85.17**(94.27)	**64.73**(**79.23**	**64.97**(**79.63**)
	2	98.23(99.77)	**69.13**(**90.60**)	**66.67**(**82.30**)	**95.27**(**95.97**)	99.17(99.00)	96.27(98.13)	**63.80**(**83.87**)	**63.83**(**83.10**)
	3	91.43(98.60)	**63.47**(**88.30**)	**66.67**(**91.40**)	**70.05**(**91.82**)	90.97(**97.43**)	**85.97**(98.00)	**66.23**(**91.20**)	**66.18**(**90.85**)
	4	77.45(99.27)	**63.22**(**88.45**)	**66.67**(**92.70**)	**69.73**(95.77)	77.88(98.90)	**70.22**(97.83)	**65.73**(**92.85**)	**65.83**(**92.75**)
O	1	91.03(97.17)	**64.77**(**74.77**)	**66.63**(**72.93**)	**72.10**(**88.47**)	89.27(96.37)	**86.57**(94.30)	**64.20**(**73.77**)	**64.13**(**73.63**)
	2	98.97(98.53)	**84.93**(**80.27**)	**76.53**(**66.33**)	**93.03** (**92.60**)	98.83(97.77)	**96.33**(97.83)	**76.43**(**62.27**)	**75.83**(**62.30**)
	3	92.35(88.08)	**57.07**(**75.67**)	**66.67**(**77.85**)	**70.53**(**79.70**)	91.75 (86.93)	**86.97**(93.08)	**67.85**(**76.87**)	**67.78**(**90.85**)
	4	95.87(88.38)	**70.73**(**69.72**)	**66.67**(**70.32**)	**78.20** (**83.22**)	**89.13**(88.17)	**88.50**(89.10)	**67.63**(**68.32**)	**67.47**(**68.23**)

1: 100(5-10), 2: 100(10-15), 3: 200(5-10), 4: 200(10-15)

sequence. However, the patterns discovered with SPM algorithms in PSAC-PDB contain 15 AA at most. Moreover, FAA patterns contained 8-10 distinct AA. Whereas the original AA sequences can contain 20 distinct AA. On three datasets (AA, AAA and ASSE), SGDT performed better than MNBT but the difference is not significant. AAA can be used more reliably for the classification and detection as compared to the AA sequences of protein structures. RF, J48 and K* performed better than NB, SVM, kNN, LR and CNN on FAA patterns discovered by using TKS and CM-SPAM. Interestingly, patterns discovered by TKS and CM-PAM are very similar to each other (see Section 3.1) but the integer-based classifiers performed better on FAA patterns of CM-SPAM compared to the FAA patterns of TKS. Majority of the classifiers performed better on 100 patterns that contain more FAA (10-15).

3.3.3. Comparison

PSAC-PDB performance was compared with recent studies for the detection and classification of SARS-CoV-2 from genome sequences. The comparison for binary classification and multi-class classification is provided in Table 14. For binary classification, the RF (highlighted in bold) results of PSAC-PDB are better than those of RF [24, 25, 31], KNN [27], SVM [30]. RF [26] performed similarly to PSAC-PDB(RF) but its performance depends on three features and their integration. The first similarity feature requires the comparison of each SARS-CoV-2 genome sequence with the RaTG13 (bat coronavirus) sequence. The other two features are based on finding the CG content in SARS-CoV-2 genome sequences. The existing methods (listed in Table 14) only performed the binary classification. PSAC-PDB can be used for both binary and multi-class classification. Some studies [28, 29] achieved 100% accuracy but their results are not included in Table 14 because their datasets contained a small number of genome sequences. For multi-class classification, RF results are included instead of MNBT because RF was faster than MNBT and achieved the overall best ACC, R, P, F1 and MCC of 99%.

Table 14: Comparison of PSAC-PDB with state-of-the-art SARS-CoV-2 detection methods.

Type	**Best Classifiers**	ACC	FPR	R	P	F1	MCC
Binary	RF [24]	0.93	–	0.93	0.93	0.93	–
	RF [25]	0.98	–	0.98	0.98	0.98	–
	RF [26]	0.99	–	0.99	0.99	0.99	0.99
	KNN [27]	0.98	–	–	–	–	–
	SVM [30]	0.97	–	0.77	0.97	0.97	–
	RF [31]	0.97	0.37	–	–	–	–
	PSAC-PDB(RF)	0.99	0.003	0.99	0.99	0.99	0.985
MC	**PSAC-PDB(RF)**	0.99	0.003	0.99	0.99	0.99	0.99

4. Conclusion

PDB plays an important role in the global dissemination of molecular structures of biological molecules. A novel framework (named PSAC-PDB) was presented in this paper, which can be used for the analysis and classification of protein structures in PDB. The framework first finds protein structures similar to a protein structure of interest. The identified similar protein structures are (1) analyzed with structural alignment and sequential pattern mining (SPM), and (2) classified by using (a) AA sequences, (b) AAA and ASSE (obtained by using a protein structures comparison tool) and (c) FAA patterns (obtained by using SPM). Three text-based and eight integer-based classifiers were used to reliably predict/classify and their performance was checked against six metrics. Two text-based classifiers (MNBT and SGDT) and three integer-based classifiers (RF, K* and J48) performed better overall. Obtained results indicated that limited (or short) AA sequences that only contain frequent sequential AA can be used to reliably predict and classify protein structures. Moreover, classifiers performed better on AAA compared to AA sequences of protein structures. PSAC-PDB achieved better results than recent approaches for SARS-CoV-2 genome sequences classification. PSAC-PDB is not limited to any particular protein structure, such as S protein of SARS-CoV-2, and can be used generally. For future, some research directions are:

- Extending the framework to (a) classify other viruses such as SARS-CoV-1, MERS, Ebola, and influenza and other protein classes. and (b) analyze and classify the S protein structures of SARS-CoV-2 that belong to various variant families such as Alpha, Delta, and Omicron.
- Considering the maximal and closed frequent patterns of AA and sequential rules between frequent AA as features for the analysis and classification. Moreover, sequence prediction models can be used to predict the next AA.
- Using alignment-free (AF) methods [59, 60] for comparison of AA sequences of protein structures, and
- Using emerging or contrast pattern mining [61] to find contrasting frequent patterns in AA for the analysis and classification of protein structures in PDB.

Conflict of Interest

Authors declare no conflict on interest.

Funding

Authors did not receive funding for this work.

Code Availability

The four datasets of AA, AAA, ASSE and FAA used for the classification experiments are available at github.com/saqibdola/PSAC-PDB.

References

[1] B. Alberts et al. "Analyzing Protein Structure and Function", In *Molecular Biology of the Cell*, 4th edition, Garland Science: New York, NY, USA, 2002.

[2] L. J. Banaszak, *Foundations of Structural Biology*, 1st Edition. Elsevier, 2000.

[3] L. Holm. "Dali server: structural unification of protein families", *Nucleic Acids Res*, 50(W1): W210-W215, 2022.

[4] N. S. A. Ghani et al. "GrAfSS: a webserver for substructure similarity searching and comparisons in the structures of proteins and RNA", *Nucleic Acids Res*, 50(W1): W375–W383, 2022

[5] L. Zhanwen et al. "FATCAT 2.0: towards a better understanding of the structural diversity of proteins", *Nucleic Acids Res*: 48(W1): W60–W64, 2020.

[6] S. Minami, K. Sawada, M. Ota M and G. Chikenji. "Mican-sq: A sequential protein structure alignment program that is applicable to monomers and all types of oligomers", *Bioinformatics*, 34(19): 3324–3331, 2018.

[7] L. Deng et al. "MADOKA: an ultra-fast approach for large-scale protein structure similarity searching", *BMC Bioinformatics*, 20(Suppl 19):662, 2019.

[8] S. Wang S, J. Ma, J. Peng and J. Xu. "Protein structure alignment beyond spatial proximity", *Scientific Reports*, 3:1448, 2013.

[9] D. Mrozek D and B. Małysiak-Mrozek B. "Cassert: A two-phase alignment algorithm for matching 3d structures of proteins", *In: Proc. International Conference on Computer Networks*, pp. 334–43, 2013.

[10] J.C. Gelly et al. “iPBA: a tool for protein structure comparison using sequence alignment strategies”, *Nucleic Acids Res*, 39 (Supp l2): W18–W23, 2011.

[11] S. B. Pandit and J. Skolnick. "Fr-tm-align: a new protein structural alignment method based on fragment alignments and the tm-score", *BMC Bioinformatics*, 9(1):531, 2008.

[12] Y. Zhang and J. Skolnick. "Tm-align: a protein structure alignment algorithm based on the tm-score", *Nucleic Acids Res*, 33(7):2302–9, 2005.

[13] J. Zhu and Z. Weng. "Fast: a novel protein structure alignment algorithm", *Proteins: Structure, Function, and Bioinformatics*, 58(3):618–27, 2005.

[14] E. Krissinel and K. Henrick. Protein structure comparison service PDBeFold at European Bioinformatics Institute. Availaele at: www.ebi.ac.uk/msd-srv/ssm

[15] G. Murzin et al. “SCOP: a structural classification of proteins database for the investigation of sequences and structures”, *Journal of Molecular Biology*, 247(4): 536–540, 1995.

[16] N. K. Fox, S. E. Brenner and J. -M. Chandonia. “SCOPe: Structural Classification of Proteins–extended, integrating SCOP and ASTRAL data and classification of new structures”, *Nucleic Acids Res*, 42(D1): D304–D309, 2014.

[17] J. -M. Chandonia, N. K. Fox and S. E. Brenner. “SCOPe: manual curation and artifact removal in the structural classification of proteins - extended database”, *Journal of Molecular Biologyl*, 429(3): 348–355, 2017.

[18] J. -M. Chandonia, N. K. Fox and S. E. Brenner. “SCOPe: classification of large macromolecular structures in the structural classification of proteins-extended database”, *Nucleic Acids Res*, 47(D1): D475–D481, 2019.

[19] J. -M. Chandonia et al. “SCOPe: improvements to the structural classification of proteins – extended database to facilitate variant interpretation and machine learning”, Nucleic Acids Res, 50(D1): D553–D559, 2022.

[20] M. Baek et al. "Accurate prediction of protein structures and interactions using a three-track neural network", *Science*, 373 (6557): 871-876, 2021.

[21] J. Jumper et al. "Highly accurate protein structure prediction with AlphaFold", *Nature*, 596: 583–589, 2021.

[22] E. W. Sayers et al. "Genbank", *Nucleic Acids Research*, 48:D84–D86, 2019.

[23] K. Kali, G. Saberwal, and G. Sharma. "The lag in sars-cov-2 genome submissions to GISAID", *Nature Biotechnology*, 39:1058–1060, 2021

[24] H. Arslan. "Machine learning methods for COVID-19 prediction using human genomic data," *Proceedings*, 74(1): 20, 2021.

[25] H. Arslan and H, Arslan. "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, 2021.

[26] H. Arslan. "COVID-19 prediction based on genome similarity of human SARS-CoV-2 and bat SARS-CoV-like coronavirus", *Computers & Industrial Engineering*, 161: 107666, 2021.

[27] Lopez-Rincon et al. "Classification and specific primer design for accurate detection of SARS-CoV-2 using deep learning," *Scientific Reports*, 11, 947, 2021.

[28] S. M. Naeem et al. "A diagnostic genomic signal processing (GSP)-based system for automatic feature analysis and detection of COVID-19", *Briefings in Bioinformatics*, 22(2):1197-1205, 2021.

[29] G. S. Randhawa. "Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study", *PLoS One*, 15(4): e0232391, 2020.

[30] I. Ahmed and G. Jeon. "Enabling artificial intelligence for genome sequence analysis of COVID-19 and alike viruses", *Interdisciplinary Sciences: Computational Life Sciences*, 14: 504-519, 2022.

[31] O. P. Singh et al. "Classification of SARS-CoV-2 and non-SARS-CoV-2 using machine learning algorithms", *Computers in Biology and Medicine*, 136:104650, 2021.

[32] S. K. Burley et al. "Protein data bank (PDB): The single global macromolecular structure archive," in *Protein Crystallography. Methods in Molecular Biology*, pp 627–641, 2017.

[33] Fournier-Viger P et al. "A survey of sequential pattern mining," *Data Science and Pattern Recognition*, vol. 1, pp. 54-77, 2017.

[34] F. Wu et al. "A new coronavirus associated with human respiratory disease in China," *Nature*, 579(7798): 265–529, 2020.

[35] D. Wrapp et al. "Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation," *Science*, 367(6483): 1260-1263, 2020.

[36] M. S. Nawaz, P. Fournier-Viger and Yulin He. "S-PDB: Analysis and classification of SARS-CoV-2 Spike protein structures," *in Proc. of BIBM*, 2022. pp. 2259-2265.

[37] L. Holm. "Using DALI for Protein Structure Comparison," *in Structural Bioinformatics. Methods in Molecular Biology*, vol 2112. Humana, New York, NY, USA, 2020.

[38] W. Kabsch and C. Sander. "Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features," *Biopolymers*, vol. 22, pp. 2577-2637, 1983.

[39] H. Cheng et al. "ECOD: An evolutionary classification of protein domains", *PLoS Computational Biology*, 10(12): e1003926, 2014.

[40] J. M. Luna, P. Fournier-Viger and S. Ventura. "Frequent itemset mining: A 25 years review," *WIREs Data Mining and Knowledge Discovery*, 9: e1329. 2019

[41] C. Zhang and S. Zhang. *Association Rule Mining, Models and Algorithms*, Springer, 2002.

[42] M. Wang, X. Shang and Z. Li. "Sequential pattern mining for protein function prediction", *in Proc. ADMA*, 2008, pp. 652-658.

[43] M. S. Nawaz et al. "Using artificial intelligence techniques for COVID-19 genome analysis," *Applied Intelligence*, 51(5): 3086-3103, 2021.

[44] M. S. Nawaz, M. Sun and P. Fournier-Viger. "Proof Guidance in PVS with Sequential Pattern Mining," *in Proc. FSEN*, 2019, pp. 45-60.

[45] P. Fournier-Viger, R. Nkambou, and E. Mephu Nguifo. "A Knowledge discovery framework for learning task models from user interactions in intelligent tutoring systems," *in Proc. MICAI*, 20008, pp. 765–778.

[46] J. M. Pokou, P. Fournier-Viger, and C. Moghrabi. "Authorship attribution using small sets of frequent part-of-speech skip-grams," *in Proc. FLAIRS Conference*, 2016, pp. 86–91.

[47] R. U. Mustafa et al. "Early detection of controversial urdu speeches from social media," *Data Science and Pattern Recognition*, 1(2): 26–42, 2017.

[48] D. Schweizer et al. "Using consumer behavior data to reduce energy consumption in smart homes: Applying machine learning to save energy without lowering comfort of inhabitants," *in Proc. ICMLA*, 2015, pp. 1123–1129.

[49] M. S. Nawaz et al. "MalSPM: Metamorphic malware behavior analysis and classification using sequential pattern mining", *Computers & Security*, 118: 102741, 2022.

[50] P. Fournier-Viger, T. Gueniche, and V. S. Tseng. "Using partially-ordered sequential rules to generate more accurate sequence prediction," *in Proc. ADMA*, 2012, pp. 431–442.

[51] S. F. Altschul et al. "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, pp. 403–410, 1990.

[52] R. Agrawal and R. Srikant. "Fast Algorithms for Mining Association Rules in Large Databases", *in Proc. VLDB*, 1994, pp. 487–499.

[53] P. Fournier-Viger et al. "TKS: Efficient mining of top-k sequential patterns," *in Proc. ADMA*, 2014, pp. 109–120.

[54] P. Fournier-Viger, A. Gomariz, M. Campos, R. Thomas. "Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information", *in Proc. PAKDD*, 2014, pp. 40-52.

[55] R. J. Urbanowicz and W. N. Browne. *Introduction to Learning Classifier Systems.* 1st Edition, Springer, 2017.

[56] X.-S. Yang. *Introduction to Algorithms for Data Mining and Machine Learning.* Elsevier; 2019

[57] P. Fournier-Viger et al. "The SPMF Open-Source Data Mining Library Version 2", *in Proc. ECML/PKDD*, 2016, pp. 36–40.

[58] E. Frank, M. A. Hall and I. H. Witten. *The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques*, Fourth Edition. Morgan Kaufmann, 2016.

[59] A. Zielezinski et al. "Benchmarking of alignment-free sequence comparison methods", *Genome Biology*, 20: 144, 2019.

[60] M. S. Nawaz et al. "COVID-19 genome analysis using alignment-free methods", *in Proc. IEA/AIE*, 2021. pp. 316–328, 2021.

[61] S. Ventura and J. M. Luna. *Supervised Descriptive Pattern Mining.* Springer, 2018.