TP-ML: A machine-learning-based tool to identify threonine proteases using sequence-derived optimal features

Ahmad Firoz, Adeel Malik, Nitin Mahajan, Le Thi Phan, Hani Mohammed Ali, Chang-Bae Kim, Balachandran Manavalan

***Abstract*—Threonine proteases (TPs) are enzymes vital for several biological processes and diseases including Alzheimer's disease and cancer. Their potential to target and degrade proteins intracellularly makes them valuable for various therapeutic and industrial applications. However, traditional experimental methods for identifying and characterizing novel TPs are exhaustive, time-consuming, and expensive. To address this, we developed TP-ML, a support vector machine-based prediction tool that can differentiate TP from non-TP sequences. We generated a benchmark dataset and calculated the physicochemical and compositional features using primary amino acid sequences. Subsequently, a comparison was made between the two feature selection approaches to identify the optimal feature sets from the original encodings. These optimal features were then used to train five different machine-learning classifiers, each assessed independently. TP-ML was selected as the best model showing consistent performance during cross-validation and independent evaluation, and achieved an accuracy of 0.934 and 0.888, respectively. We anticipate TP-ML to be a powerful tool for identifying TPs, aiding in their experimental characterization and industrial application exploration. TP-ML predictor is freely accessible at https://procarb.org/TP-ML/.**

***Index Terms*—Threonine proteases, Support vector machine, Feature selection, Boruta, Recursive feature elimination.**

This research was funded by the Institutional Fund Projects under Grant No. (IFPIP:350-130-1443). The authors gratefully acknowledge the technical and financial support provided by the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia. *(Corresponding authors: Chang-Bae Kim and Balachandran Manavalan)*. Ahmad Firoz and Adeel Malik have contributed equally and are co-first authors.

Ahmad Firoz is with Department of Biological Sciences, Faculty of Science, King Abdulaziz University, Jeddah, Saudi Arabia, and also with Princess Dr. Najla Bint Saud Al-Saud Center for Excellence Research in Biotechnology, King Abdulaziz University, Jeddah, Saudi Arabia (email: aakram@kau.edu.sa).

Adeel Malik is with Institute of Intelligence Informatics Technology, Sangmyung University, Seoul, 03016, Republic of Korea (email: adeel@procarb.org).

Nitin Mahajan is with Wugen, St. Louis, MO 63110, USA (email: dr.nitin20@yahoo.com).

Le Thi Phan is with Computational Biology and Bioinformatics Laboratory, Department of Integrative Biotechnology, College of Biotechnology and Bioengineering, Sungkyunkwan University, Suwon, 16419, Gyeonggi-do, Republic of Korea (email: phanthile@g.skku.edu).

Hani Mohammed Ali is with Department of Biological Sciences, Faculty of Science, King Abdulaziz University, Jeddah, Saudi Arabia, and also with Princess Dr. Najla Bint Saud Al-Saud Center for Excellence Research in Biotechnology, King Abdulaziz University, Jeddah, Saudi Arabia (email: hmohammedali@kau.edu.sa).

Chang-Bae Kim is with Department of Biotechnology, Sangmyung University, Seoul, 03016, Republic of Korea (email: evode-vo@smu.ac.kr).

Balachandran Manavalan is with Computational Biology and Bioinformatics Laboratory, Department of Integrative Biotechnology, College of Biotechnology and Bioengineering, Sungkyunkwan University, Suwon, 16419, Gyeonggi-do, Republic of Korea (email: bala2022@skku.edu).

I. Introduction

PROTEOLYSIS is one of the most important biological reactions, facilitating the degradation of proteins into smaller polypeptides or amino acids (AAs) via cellular enzymes called proteases (also known as proteolytic enzymes and peptidases). These enzymes catalyze the hydrolysis of peptide bonds by targeting the carbonyl group of the peptide [1], [2]. They are ubiquitously distributed in cellular compartments to perform significant biological processes. Genomic analysis revealed the presence of >900 protease genes and >1600 protease inhibitory genes in human and mouse genomes, underscoring the critical role of proteolysis in regulating cellular function [3]. Proteases specifically cleave proteins either from the N-terminal (aminopeptidases) or C-terminal (carboxypeptidases) and in the middle of the molecule (endopeptidases) [4]. Based on the nature of the AA present in the active site of the enzyme and the mechanism of the peptide bond cleavage, proteases are categorized as aspartate (Asp), cysteine (Cys), glutamic acid (Glu), serine (Ser), threonine (Thr), proteases, as well as matrix metalloproteases [5]. For Cys, Ser, and Thr proteases, the key AA residue acts as a nucleophile, whereas in others, key AAs trigger a water molecule which then works as a nucleophile [6], [7]. Proteases are imperative in diverse key processes such as cell cycle progression, cell proliferation, DNA replication, tissue remodeling, hemostasis, and the immune response [8].

Threonine proteases (TPs), a class of endopeptidases, are contained within clan PB as they utilize the N-terminal threonine as the active site. They are commonly found in bacteria, yeast, and other living organisms. The prototype members of TP enzymes are the catalytic subunits of eukaryotic proteasomes and are an essential part of the protein turnover system [9]. According to the peptide database [10], TPs are classified into 6 families: T1 (subfamilies T1A and T1B), T2, T3, T5, T7, and T8. TPs use a catalytic charge relay system to activate secondary hydroxyl nucleophiles for catalysis. The active site of these proteases contains a unique threonine residue that acts as a nucleophile to cleave peptide bonds in substrates. The mechanism of catalysis is a two-step process: (i) formation of a covalent acyl-enzyme intermediate as a result of a nucleophile attack on the substrate and (ii) hydrolysis of the intermediate to revive the free enzyme and emancipation of the product

[11]. Therefore, these sites play an essential role in multiple catalytic subunits of the proteasome. TPs degrade cellular proteins, which are tagged through a complex modification (i.e., polyubiquitination). As the name suggests, polyubiquitination adds a series of ubiquitin molecules to the protein targeted for degradation. The expression of most proteins is controlled by the ubiquitin-proteasome system, which includes ubiquitin, the E1, E2, and E3 ubiquitin ligase machinery, and deubiquitinating enzymes. Proteosome assembles into a large complex to position its substrates and uses a Thr-Glu/Asp-Lys triad. The enzyme structure can be capped (26S proteasome) or uncapped (20S proteasome). The 20S proteasome core is made up of four rings attached to seven different proteins. The outer rings are composed of two α rings (α1–7 subunits), and two inner rings are composed of β rings (β1–7 subunits). Four out of seven β subunits are catalytically inactive (β3, β4, β6, and β7). The catalytically active β subunits (β1, β2, β5) are stacked into the basic 20S core [11]–[13]. It is important to note that each subunit is encoded by different genes, thereby altering the proteasome's catalytic properties and diversity. The tertiary structure shows the presence of an alpha/beta/beta/alpha sandwich, where the beta-sheet has four strands and an active site. This structure has been demonstrated like the Ntn hydrolases in the PB clan which include penicillin acylase and glycosylasparaginase. Various posttranslational modifications, including phosphorylation, N-acetylation, glycosylation, ubiquitination, and more have been observed for these protein subunits [14].

Apart from proteasome, Testes-specific protease 50 (TSP50) is another important member of TPs which shares protein sequences and structures with many Ser proteases [15]. Physiologically, TSP50 is expressed in spermatocytes, while various reports have shown the high expression in more than 90% of breast cancer, colorectal cancer, and cervical and gastric cancer tissues [16]–[18]. The catalytic triad of TSP50, especially Thr310, plays a critical role in its protease activity [15], [19]. Li et al (2012) have demonstrated that T310A mutation in TSP50 impairs its ability to promote cell proliferation, colony formation, and tumorigenicity [15]. In addition, TSP50 gene locus 3p21.31 has been demonstrated as a susceptible locus for colorectal cancer in the Chinese population [20].

Under diseased conditions, endogenous inhibitor checkpoints fail, resulting in a skewed protease-antiprotease balance. Overexpression or dysregulated activity of proteases is demonstrated to play an important role in various diseases like high blood pressure, diabetes, infections, and cancer. Considering their inevitable role in various diseases, this class of enzymes can be exploited as potential drug targets. Usage of several proteases and inhibitors have been applied in various food and drug industries. Proteases account for 60% of the total enzyme market including 25 US Food and Drug Administration (FDA)-approved products [21], [22]. Bortezomib, a synthetic compound, is the first US FDA-approved proteasome inhibitor used as therapy for multiple myeloma (MM) and relapsed/refractory MM and mantle cell lymphoma. Another emerging application of TPs is in biocatalysis, where they can be used to cleave specific bonds in complex molecules and produce modified proteins with specific biological activities [7], [23].

In this study, we introduce TP-ML, the first machine-learning (ML)-based tool designed to predict TPs utilizing sequence-based optimal features (Figure 1). The development of such a computational method is imperative as traditional experimental techniques are laborious, costly, and time-consuming. These computational approaches offer alternative strategies for the rapid identification and annotation of novel sequences. Currently, available computational tools like BLAST and HMMER can support identifying sequences, but their effectiveness is restrained to cases where there is a sufficient degree of similarity between the target and query sequences. Consequently, ML-based methods are considered robust substitutes for such classification tasks. We anticipate that TP-ML will serve as a valuable and efficient tool to identify TPs, thereby assisting the exploration of their industrial and functional applications. The TP-ML is freely accessible at https://procarb.org/TP-ML/.

II. METHODS

A. Dataset acquisition

The protein sequence datasets were downloaded from the MEROPS version 12.4 database [10]. The members of TPs constitute the positive dataset whereas the remaining families of proteolytic enzymes constitute the negative dataset (non-TP proteins). Since the length of sequences in positive data is in the range of 100-1000 AAs, we selected negative dataset sequences within the same length range. After merging both datasets we removed redundant sequences using CD-HIT version 4.8.1 [24]. All sequences exhibiting >40% sequence identity were excluded from the analysis. Redundant sequence removal resulted in an imbalanced dataset with 685 positive and approximately 41,000 negative sequences. To generate a balanced dataset, 750 negative sequences were randomly selected from the non-TP proteins. The final dataset comprises 685 positive and 750 negative sequences, which were subsequently divided into training (1005 sequences) and independent validation (430 sequences) sets.

B. Feature encodings

We employed ten different sequence-derived feature encodings to train the ML classifiers. The 'protr' package [25] was used to transform various length sequences into fixed length feature vectors. These features represent the key characteristics of a protein sequence as detailed below:

1) Amino acid composition (AAC): The AAC of a protein sequence is defined as a count of each AA for each TP and non-TP protein sequence, normalized by the total length of the protein sequence [26]. AAC has a fixed length of 20D feature vector, and mathematically it is expressed as:

$$AAC(i) = \frac{R_i}{L} \tag{1}$$

where R_i represents the number of AAs of type i, and L denotes protein sequence length.

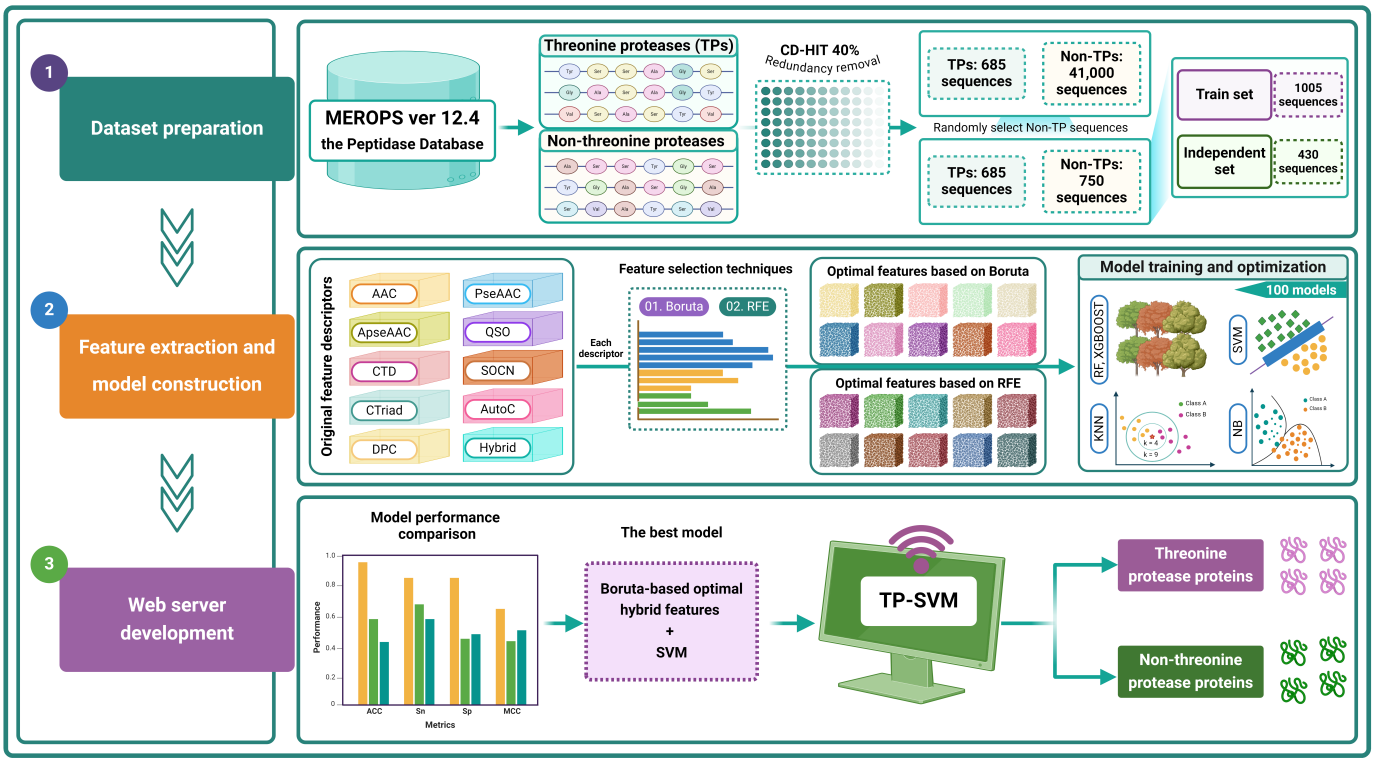


Fig. 1: Overview of TP-ML. It involves the following steps: (1) Dataset preparation; (2) Feature extraction and model construction; (3) Final model selection and webserver development.

2) Pseudo-amino acid composition (PseAAC): PseAAC descriptors are also known as the type 1 pseudo-AA composition and were first proposed by Chou [27] to predict membrane protein type and protein subcellular localization. PseAAC represents a blend of a set of separate sequence correlation factors and the 20 elements of the traditional AAC. In general, these complementary factors are a series of rank different correlation factors along a protein sequence [27], [28]. However, they can also represent any fusion of other components as long as they can exhibit some kind of sequence order effects.

3) Amphiphilic pseudo-amino acid composition (APseAAC): APseAAC is also known as the type 2 pseudo-AA composition and was also first proposed by Chou [28] to predict enzyme subfamily classes. APseAAC features include 20 + 2λ discrete numbers. Among these, the first 20 also represent the elements of the typical AA frequency, and the next 2λ numbers represent a set of correlation factors that mirror distinct hydrophobicity and hydrophilicity distribution arrangements along a protein sequence.

4) Autocorrelation (AutoC): The AutoC encodings collect information regarding the physicochemical properties of a protein chain, which may improve the performance of an ML model [29]. The AutoC descriptors can be broadly grouped into three categories:

(i) Moran AutoC encodings:

$$AC(d) = \sum_{i=1}^{N-d} P_i P_{i+d}, \quad d = 1, 2, \ldots, \text{nlag} \tag{2}$$

where d is the autocorrelation lag; P_i and P_{i+d} are the AA properties at position i and i+d; *nlag* represents the maximum value of the lag.

(ii) Moreau-Broto AutoC descriptors:

$$I(d) = \frac{\frac{1}{N-d}\sum_{i=1}^{N-d}(P_i - \overline{P})(P_{i+d} - \overline{P})}{\frac{1}{N}\sum_{i=1}^{N}(P_i - \overline{P})^2}, \quad d = 1, 2, \ldots, 30 \tag{3}$$

(iii) Geary AutoC descriptors:

$$C(d) = \frac{\frac{1}{2(N-d)}\sum_{i=1}^{N-d}(P_i - P_{i+d})^2}{\frac{1}{N}\sum_{i=1}^{N}(P_i - \overline{P})^2}, \quad d = 1, 2, \ldots, 30 \tag{4}$$

$\overline{P}$ is the average value of property P denoted as $\overline{P} = \frac{\sum_{i=1}^{N} P_i}{N}$.

5) Composition (C), Transition (T), and Distribution (D) (CTD): Since their first application in the protein folding classes [30], [31], CTD descriptors have been exploited in several bioinformatics applications [32]–[38]. The 20 naturally occurring AAs in CTD descriptors are divided into three groups (polar, neutral, and hydrophobicity) based on seven distinct physicochemical properties such as solvent accessibility, polarizability, polarity, hydrophobicity, charge, normalized van der Waals volume, and secondary structure (Table S1).

In CTD, C corresponds to the percentage composition of a target protein sequence's hydrophobic, neutral, and polar residues. It can be mathematically expressed as:

$$C(a) = \frac{Z_a}{K}, \quad a \in \{\text{neutral}, \text{polar}, \text{hydrophobic}\} \tag{5}$$

where Z_a is the number of AAs of type a in the given sequence.

In CTD, T comprises three values (hydrophobic, neutral, and polar). A transition from a neutral group to a hydrophobic group is the frequency with which a hydrophobic residue is followed by a neutral residue or vice versa. The transitions between neutral and polar groups, and polar and hydrophobic groups, are also described similarly. Mathematically, T can be defined as follows:

$$T(ab) = \frac{Z_{ab} + Z_{ba}}{K - 1}, \tag{6}$$

$a, b \in \{(\text{polar}, \text{neutral}), (\text{neutral}, \text{hydrophobic}), (\text{hydrophobic}, \text{polar})\}$
Z_{ab} and Z_{ba} represent the number of dipeptides encoded as *ab* and *ba* in the sequences.

Finally, D in CTD comprises five values for each of the three classes and estimates the percentage of a target protein sequence length within which residues about an explicit attribute are found within 25, 50, 75, and 100% of their position. Overall, CTD generates 147-dimensional (D) features (21 x 7), and each physico-chemical property is characterized by a 21D feature vector.

6) Conjoint triad (CTriad): Shen et al. [39] first exploited CTriad encodings for predicting protein-protein interactions. For any target protein sequence, CTriad encodings represent a vector space containing descriptors of AAs. The 20 AAs are clustered based on their side chain volumes and dipoles to reduce the vector space. Eventually, in CTriad encoding, a 343D feature vector is generated for a target protein chain. CTriad encodings can be expressed mathematically as:

$$d_i = \frac{f_i - \min\{f_1, f_2, \ldots, f_{343}\}}{\max\{f_1, f_2, \ldots, f_{343}\} - \min\{f_1, f_2, \ldots, f_{343}\}}, \quad i = 1, 2, \ldots, 343 \tag{7}$$

where f_i (*i*=1, 2, 3, ..., 343) is the frequency of occurrence of each triad.

7) Dipeptide composition (DPC): DPC is calculated as the frequency of any two naturally occurring AAs observed in a protein chain. In each sequence, there are 20 x 20 combinations of AA pairs giving rise to a 400D feature vector [40]. Mathematically, DPC can be defined as:

$$DPC(ab) = \frac{z_{ab}}{K - 1} \tag{8}$$

where Z_{ab} represents the number of dipeptides encoded as *ab* in a given sequence, and *K* is the protein sequence length.

8) Quasi-sequence order (QSO): Recognizing the fact that several numbers of sequence order patterns exist in biological sequences, it is impractical to add this information directly into an ML classifier [41], [42]. Therefore, to address this issue, QSO is exploited to indirectly incorporate the sequence order information [43]. QSO encodings are estimated by using the Grantham distance matrix and Schneider-Wrede distance matrix for each pair of 20 naturally occurring AAs [25], [44]. The former contains information about the chemical distance, whereas the latter matrix computes the physicochemical properties such as polarity, hydrophobicity, and hydrophilicity [45].

9) Sequence order coupling number (SOCN): The dth rank sequence-order-coupling number is defined as:

$$\tau_d = \sum_{i=1}^{N-d} \left(d_{i,i+d}\right)^2, \quad d = 1, 2, \ldots, \text{maxlag} \tag{9}$$

where $d_{i,i+d}$ is the entry in a given distance matrix representing the distance between the amino acids at position i and $i + d$, *maxlag* exhibits the maximum value of the lag, and N is the length of the protein sequence. It should be noted that the length of the protein sequence must not be less than *maxlag*.

10) Hybrid encodings: This encoding was created by combining all the nine features listed in section II (B1-B9). The hybrid encodings form a 1920D feature vector which was also used for training different ML classifiers.

C. Machine-learning classifiers

In the present study, we tested five ML classifiers using the caret package in R [46].

1) K-nearest neighbor (KNN): KNN is one of the simplest and fastest types of ML classifiers and can be utilized for both supervised and unsupervised learning [47]. This approach aims to find k nearest neighbors in a dataset when compared to a new sample [48]. Distances between examples are computed for each feature using a distance metric like Euclidean, Manhattan, or Mahalanobis [49]. The only hyper-parameter available for tuning in KNN is the value of k itself. Determining a reasonable equilibrium between underfitting and overfitting depends on the optimal value of the k parameter [50]. A smaller value of k can lead to noise points and if the k value is too large, the neighborhood may include points from other classes. One of the key advantages of KNN is that there is no cost associated with the learning process.

2) Naïve Bayes (NB): NB is a type of probabilistic classification algorithm based on Bayes' theorem which assumes that the presence of a specific feature in a class is independent of the presence of any other descriptor [51]. This conditional independence assures that how one descriptor influences a result in no way interacts with how another variable impacts the analogous outcome [52]. In NB, the penultimate result is the outcome of the impartial contribution from each feature [53]. NB streamlines predictive modeling issues to escape the curse of dimensionality [52]. NB classifier can be best exploited for bigger datasets which may have millions of data samples or images.

3) Random Forest (RF): RF was first developed by Breiman [54], and since then it has been a widely used ensemble learner that can handle both classification as well as regression problems. Compared to other classifiers, RF is considered superior due to its ease of training, fast prediction abilities, and interpretability [55]. RF is an ensemble technique encompassing several decision trees. In each tree, "n" descriptors are selected randomly from the entire feature set [56]. This trait of random selection makes RF unbiased and decreases the correlation between the unpruned trees [57]. Initially, a bagging algorithm creates a training feature set using resampled instances. Thereafter, a decision tree is generated using a randomly selected feature vector, and the resampled training set [58], [59]. Finally, the predictions of all the decision trees are compiled, and the final prediction is determined by majority voting [60].

4) Support Vector Machine (SVM): SVM comprises a set of supervised learning techniques commonly employed for both regression and classification tasks [61]. SVM is based on the principle of decision planes that use decision boundaries to ideally split data into distinct classes [62]. SVMs employ statistical learning theory and the principle of structural risk minimization to optimize generalization performance [43]. In SVM, the training data is grouped into two classes, and the algorithm maximizes the distance between these classes and the hyperplanes [63]. By leveraging ML theory, SVMs can prevent overfitting and enhance prediction accuracy. Consequently, SVMs often outperform other classifiers [64], [65].

5) eXtreme Gradient Boosting (XGBOOST): XGBOOST is one of the leading ML approaches, known for its high speed and performance as a precursor to gradient-boosted decision trees [47], [66]. While the performance and speed of XGBOOST-based models can be enhanced by integrating novel features into gradient trees [67], exploiting XGBOOST to enhance performance is not simple. This sophistication emerges from the several parameters involved in the classifier, making hyper-parameter optimization both challenging and indispensable for enhancing model performance [53].

The performance of each of the above-mentioned classifiers was optimized by fine-tuning essential hyper-parameters. A grid-based search together with a 10-fold cross-validation (CV) was applied to assess the influence of each hyper-parameter (Table S2). Overall, the classifier exhibiting the best performance was selected.

D. Feature Selection

Selection of an optimal feature subset is desirable as all features do not contribute equally to a robust ML-based predictor [68]. In the present study, we used the R implementation of two feature selection techniques i.e. recursive feature elimination [69] and Boruta (v7.0.0) [70]. Table S3 provides the overview of the dimension size of all 10 original feature encodings and dimension size after applying feature selection techniques.

1) Recursive Feature Elimination (RFE): RFE works on a backward selection protocol and avoids refitting numerous models at individual search steps [69]. This selection process repeatedly removes the least significant feature until a specific subset of encodings is identified. We tested RFE on all 10 feature encodings using all five ML models. During the RFE cycle, several subsets of the training data with variable dimension sizes were created. These subsets were used as input vectors for an RF-based classifier using a 10-fold CV.

2) Boruta: In the Boruta algorithm, features are selected and ranked based on feature importance score for predictors [70]. It creates copies of the real feature by randomly shuffling other features known as shadow features. The relevance of the real predictor is statistically compared to the highest significance score of shadow variables and labeled as relevant or irrelevant based on these scores. All irrelevant shadow variables are subsequently eliminated. This step is repeated till all variables are labeled as either relevant or irrelevant or for a predefined number of iterations (maxRuns) [71]. Here in the present study, we selected the maxRuns = 1000. Altogether, Boruta follows a top-down procedure for suitable variables by comparing them with an original set of features.

III. Results

A. Summary of the dataset

The final non-redundant dataset consists of 1435 sequences including 685 TP (positive) and 750 randomly selected non-TP (negative) proteins retrieved from the MEROPS database. To compare the over-representation of various COG categories in the positive and negative datasets, we used EggNOG mapper for their functional annotation [72]. A significant difference in the few enriched categories was observed between the two datasets (Figure 2). A major proportion ($\sim$59%) of TP sequences belonged to the COG category AA transport and metabolism (E). In contrast, cell wall/membrane/envelope biogenesis (M), lipid transport and metabolism (I), and defense mechanisms (V) were the top over-represented annotated categories in the negative dataset. About 25% of sequences in both datasets belonged to the post-translational modification, protein turnover, and chaperone functions (O) category. Moreover, the number of proteins with unknown functions (S) and proteins with no hits in the COG database (NA) were significantly higher in the negative dataset.

B. Compositional differences between TP and non-TP sequences

To examine any compositional differences between TP and non-TP enzymes, we compared the AAC of both these datasets. Figure 3 illustrates that, of the 20 AAs, both TPs and non-TPs are enriched with at least seven statistically significant AAs (Wilcoxon test; $p < 0.05$). Among the seven most dominant AAs in TPs, three are represented by non-polar and aliphatic R groups containing residues including alanine (A), glycine (G), and valine (V). The other four residues include two polar uncharged AAs (methionine (M) and threonine (T)), a positively charged arginine (R), and

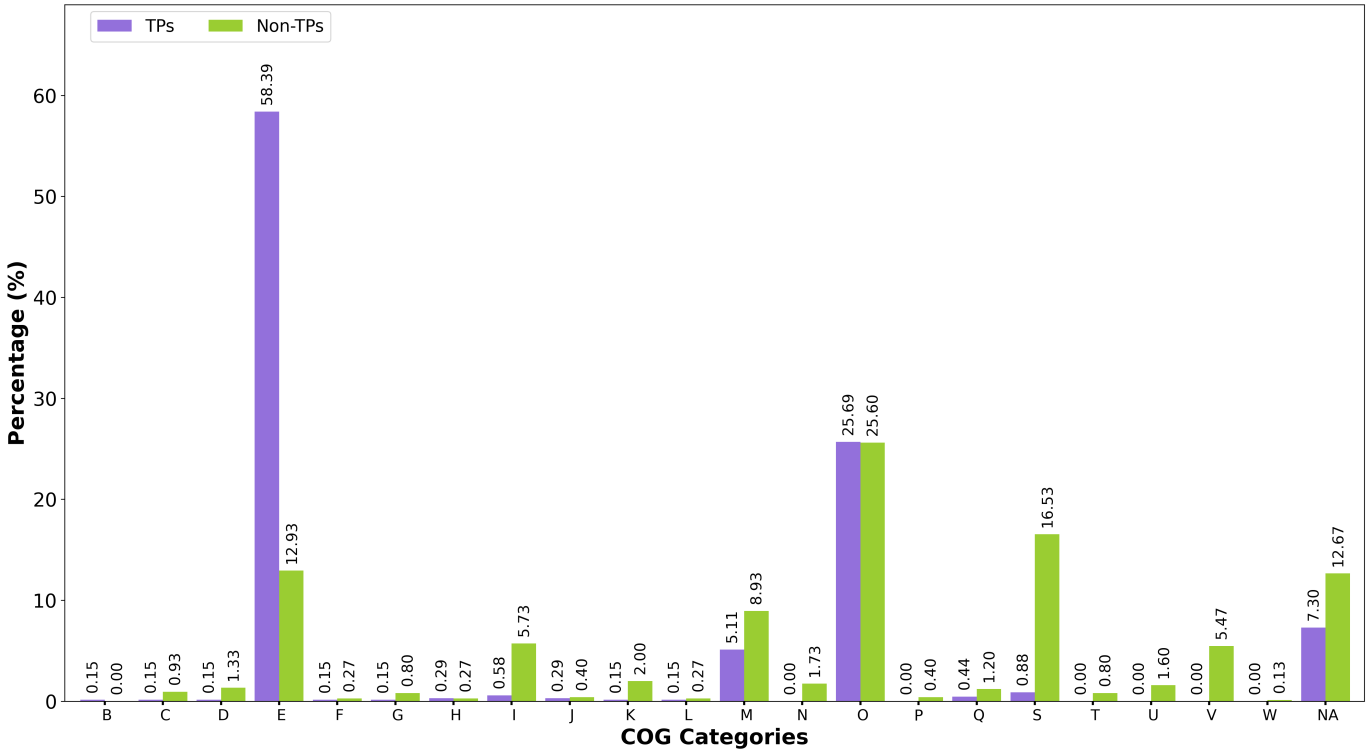


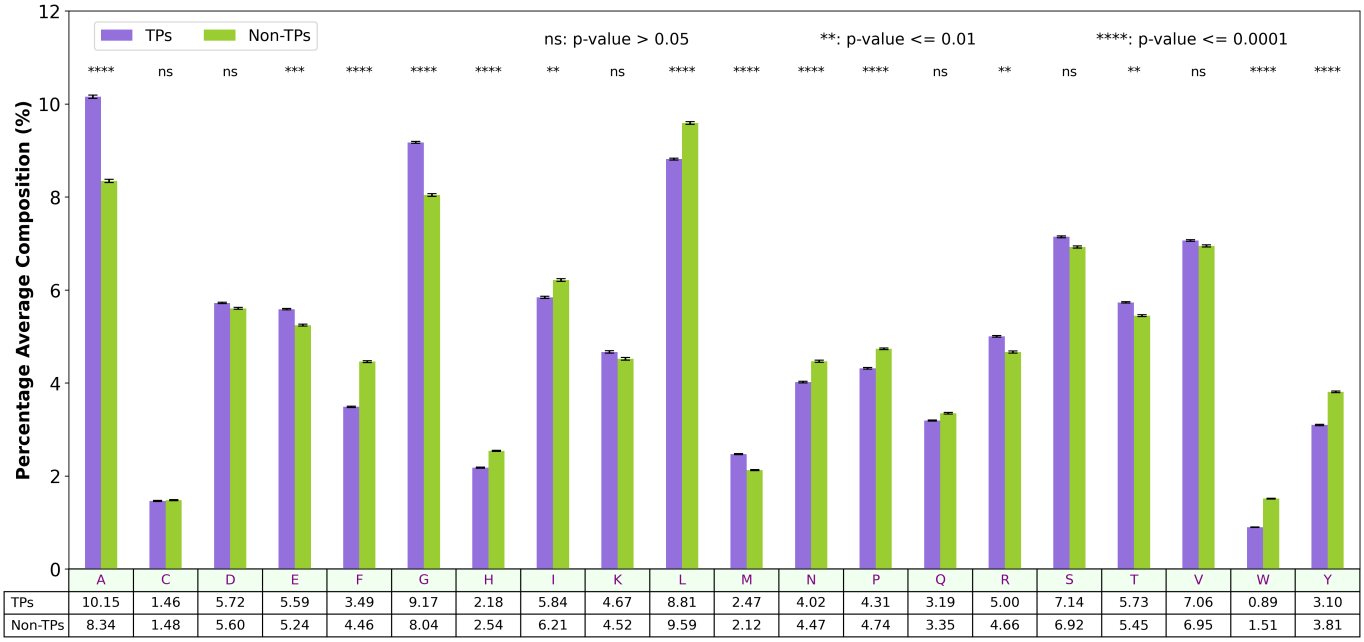
Fig. 2: Distribution of various COG categories in TP and non-TP datasets. The COG categories on the x-axis represent Chromatin structure and dynamics (B); Energy production and conversion (C); Cell cycle control, cell division, and chromosome partitioning (D); AA transport and metabolism (E); Nucleotide transport and metabolism (F); Carbohydrate transport and metabolism (G); Coenzyme transport and metabolism (H); Lipid transport and metabolism (I); Translation, ribosomal structure, and biogenesis (J); Transcription (K); Replication, recombination, and repair (L); Cell wall/membrane/envelope biogenesis (M); Cell Motility (N); Posttranslational modification, protein turnover, and chaperones (O); Inorganic ion transport and metabolism (P); Secondary metabolites biosynthesis, transport, and catabolism (Q); Function unknown (S); Signal transduction mechanisms (T); Intracellular trafficking, secretion, and vesicular transport (U); Defense mechanisms (V); Extracellular structures (W); and Not applicable (NA).

a negatively charged glutamic acid (E). In contrast, non-TP sequences exhibited the dominance of residues with aromatic groups including phenylalanine (F), tyrosine (Y), and tryptophan (W). Additionally, two non-polar and aliphatic R groups containing residues, leucine (L) and proline (P) were also over-represented in non-TPs. Among all non-polar aliphatic AAs, proline is unique due to its five-membered ring structure. Asparagine (N), a polar uncharged AA, and histidine (H), a positively charged AA, also exhibited dominance in non-TP sequences. Therefore, a significantly lower frequency of aromatic residues in TPs could be one of the most important characteristic features for the classification of TPs from non-TPs. These unique compositional disparities imply that our model may exploit the existence of distinct AAs as a feasible way to differentiate TP from non-TP sequences.

C. Comparison of different ML classifiers using Boruta-based optimal features

We investigated the impact of ten distinct feature encodings: AAC, APseAAC, AutoC, CTD, CTriad, DPC, PseAAC, QSO, SOCN, and hybrid features, in conjunction with five different ML classifiers (KNN, NB, RF, SVM, and XGBOOST) for distinguishing TP from non-TP sequences. Specifically, we employed the Boruta technique to eliminate irrelevant features for each descriptor, resulting in ten Boruta-derived optimal feature encodings with optimal feature dimensions (see Table S3). These optimal descriptors were then input into the five classifiers, producing 50 models (10 Boruta-based optimal encodings $\times$ 5 ML classifiers) using an extensive search range (refer to Table S2) and a 10-fold CV strategy (Figures 4A-J). The top three models, employing hybrid-based encodings and trained with RF, SVM, and XGBOOST, achieved an ACC of $\geq 90\%$. Among these, the SVM-based model demonstrated the highest MCC of 0.869, while the RF and XGBOOST models achieved an MCC of approximately 0.800. The remaining 47 models exhibited ACC and MCC ranges of 0.576–0.889 and 0.298–0.782, respectively. The model utilizing DPC-based optimal encodings with the KNN classifier demonstrated the poorest performance (Figure 4F).

Next, we estimated the performance of each ML classifier irrespective of the 10 optimal feature encodings in predicting TP sequences. The result shows that SVM performs best among the other four classifiers in terms of both ACC and MCC (Table I). Notably, the average ACC of the SVM



	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
TPs	10.15	1.46	5.72	5.59	3.49	9.17	2.18	5.84	4.67	8.81	2.47	4.02	4.31	3.19	5.00	7.14	5.73	7.06	0.89	3.10
Non-TPs	8.34	1.48	5.60	5.24	4.46	8.04	2.54	6.21	4.52	9.59	2.12	4.47	4.74	3.35	4.66	6.92	5.45	6.95	1.51	3.81

Fig. 3: Differences in amino acid composition between TP and non-TP sequences.

TABLE I: Performance metrics for different feature selection techniques. Values in parentheses represent the standard deviations.

*FST	Metrics	KNN	NB	RF	SVM	XGBOOST
Boruta	Accuracy	0.730 (0.080)	0.725 (0.047)	0.828 (0.053)	0.847 (0.048)	0.826 (0.048)
	Sensitivity	0.891 (0.091)	0.745 (0.138)	0.778 (0.056)	0.819 (0.053)	0.809 (0.049)
	Specificity	0.582 (0.176)	0.707 (0.103)	0.873 (0.052)	0.873 (0.046)	0.842 (0.051)
	MCC	0.503 (0.128)	0.461 (0.097)	0.656 (0.106)	0.693 (0.096)	0.652 (0.097)
RFE	Accuracy	0.701 (0.096)	0.713 (0.059)	0.824 (0.049)	0.797 (0.110)	0.840 (0.041)
	Sensitivity	0.895 (0.095)	0.738 (0.143)	0.761 (0.054)	0.696 (0.262)	0.826 (0.039)
	Specificity	0.522 (0.229)	0.690 (0.154)	0.881 (0.050)	0.888 (0.050)	0.852 (0.047)
	MCC	0.451 (0.156)	0.440 (0.105)	0.649 (0.100)	0.656 (0.110)	0.679 (0.083)

*FST, feature selection technique.

classifier is about 2–12% higher compared to the other four classifiers. Likewise, the corresponding average MCC of the SVM-based classifier is approximately 4–23% higher than KNN, NB, RF, and XGBOOST-based classifiers.

Additionally, independent of the classifier used, we also determined the best-performing feature descriptor to classify TP sequences. The analysis suggests that hybrid-based optimal features are the top encodings and achieved an average ACC of 0.886 and MCC of 0.773, followed by CTD which achieved an ACC of 82% and MCC of 0.643 (Table II). The performance of other encodings in terms of ACC varied between 72–80%.

D. Comparison of different ML classifiers using RFE-based optimal features

In addition to the Boruta method, we employed the RFE technique to select optimal features for each feature descriptor across 10 feature encodings, subsequently inputting each into five ML classifiers as previously mentioned. This approach also resulted in the generation of 50 models (10 RFE-based optimal encodings $\times$ 5 ML classifiers), which were evaluated using a 10-fold CV. Figures 5A-J illustrate the performance of each feature descriptor trained with the five ML classifiers. The results indicate that the top-performing model utilized XGBOOST with RFE-based optimal hybrid encodings, achieving an ACC of 0.905 and an MCC of 0.811. Notably, it was the only model to achieve an ACC above 90% under this feature selection technique. However, its ACC and MCC are 2.89% and 5.80% lower than the best model under the Boruta technique. In addition, at least three models under the Boruta method exhibited an ACC $\geq$90% (Figure 4J), whereas only one model under the RFE method achieved an ACC $\geq$90% (Figure 5J). The remaining 49 models exhibited performance in terms of ACC ranging from 0.489 to 0.882, with corresponding MCC scores between 0.101 and 0.763. Consistent with the findings from the previous section, the DPC-based optimal features with the KNN classifier once again exhibited the worst-performing model when utilizing the RFE technique (Figure 5F).

We then compared the performance of five ML classifiers irrespective of the 10 optimal feature encodings in classifying TP from non-TP sequences. From Table I, we observe that XGBOOST outperforms the other four classifiers in both ACC and MCC when RFE-based optimal encodings are exploited.

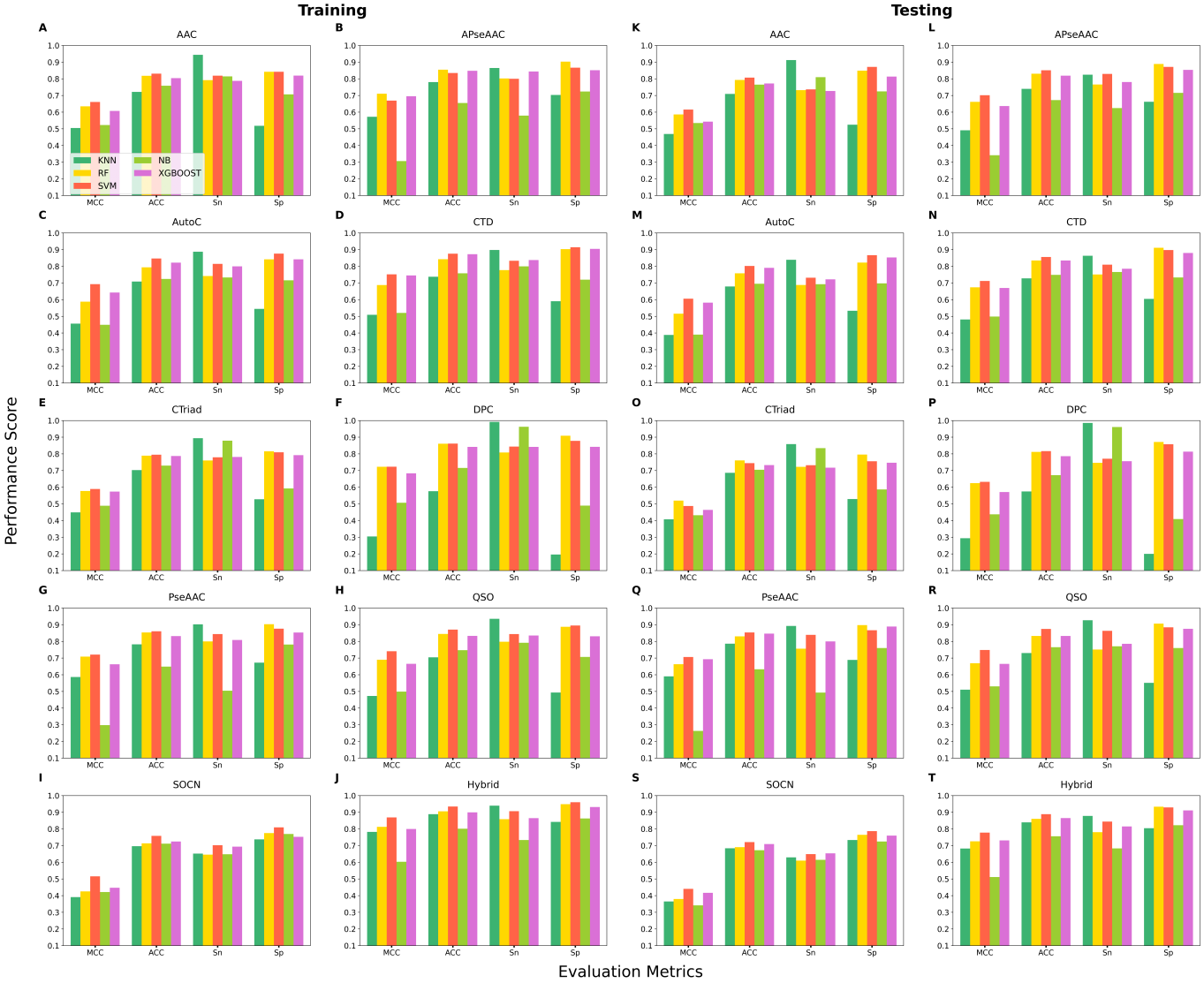


Fig. 4: Performance comparison of five ML classifiers and Boruta-based optimal feature encodings on CV (panels A to J) and independent validation (panels K to T) for AAC, APseAAC, AutoC, CTD, CTriad, DPC, PseAAC, QSO, SOCN, and Hybrid feature descriptors, respectively.

However, the average ACC achieved by XGBOOST is slightly lower than the average ACC displayed by the Boruta-based SVM model. Additionally, the Boruta-based SVM model exhibited a 2% higher MCC value than the average MCC achieved by RFE-based XGBOOST models.

Next, when we attempted to determine the best-performing RFE-based optimal encodings for the prediction of TP and non-TP sequences independent of the classifier. We observe that four encodings, namely, QSO, CTD, ApseAAC, and PseAAC achieved an average ACC of ≥ 0.800 (Table III). This is about 8% lower than the average ACC exhibited by Boruta-based hybrid encodings. The results suggest that Boruta-based optimal hybrid features can better differentiate between TP and non-TP sequences.

E. Comparison of ML classifiers using two different optimal feature sets on an independent validation dataset (IVDS)

IVDS was used to assess the performance of 50 models using optimal encodings derived from the Boruta algorithm (Figures 4K-T). The results of independent validation and 10-fold CV exhibited consistency in the ACC scores of these two datasets when Boruta-based optimal feature encodings were utilized. For example, the SVM-based model using optimal hybrid encodings was the top performer achieving the best ACC of 0.934, and 0.888 during training and independent validation, respectively (Figures 4J, 4T). Similarly, the KNN classifier based on optimal DPC encodings which ranked last in the training stage also exhibited the worst performance during independent validation. In addition to the similarities, some differences were also observed. The second-best performing model during 10-fold CV was based on RF and optimal hybrid encodings with an ACC of 0.905. However, this model ranked

TABLE II: Comparison of 10 Boruta-based optimal encodings on five different ML classifiers. Values in parentheses represent the standard deviations.

Feature	ACC	Sn	Sp	MCC
AAC	0.786 (0.046)	0.831 (0.064)	0.746 (0.139)	0.586 (0.069)
APseAAC	0.794 (0.083)	0.778 (0.114)	0.810 (0.090)	0.591 (0.168)
AutoC	0.779 (0.060)	0.795 (0.137)	0.764 (0.137)	0.566 (0.110)
CTD	0.817 (0.065)	0.829 (0.046)	0.806 (0.146)	0.643 (0.120)
CTriad	0.761 (0.042)	0.819 (0.063)	0.707 (0.137)	0.535 (0.063)
DPC	0.771 (0.125)	0.890 (0.082)	0.663 (0.311)	0.588 (0.182)
PseAAC	0.795 (0.880)	0.772 (0.155)	0.817 (0.093)	0.595 (0.175)
QSO	0.800 (0.071)	0.841 (0.058)	0.763 (0.168)	0.614 (0.120)
SOCN	0.721 (0.023)	0.668 (0.027)	0.769 (0.027)	0.440 (0.047)
Hybrid	0.886 (0.050)	0.860 (0.078)	0.909 (0.053)	0.773 (0.101)

TABLE III: Comparison of 10 RFE-based optimal encodings on five different ML classifiers.

Features	ACC	Sn	Sp	MCC
AAC	0.786	0.831	0.746	0.586
APseAAC	0.809	0.789	0.827	0.621
AutoC	0.780	0.795	0.767	0.574
CTD	0.813	0.831	0.796	0.635
CTriad	0.693	0.780	0.613	0.425
DPC	0.733	0.839	0.637	0.506
PseAAC	0.807	0.779	0.833	0.617
QSO	0.813	0.842	0.786	0.636
SOCN	0.720	0.662	0.773	0.438
Hybrid	0.768	0.653	0.874	0.664

4^{th} on IVDS. In contrast, the 2^{nd} top-performing model used SVM and optimal QSO-based features during independent validation. However, this model ranked 7^{th} on the training dataset.

Similarly, when we compared the performance of all 50 models using RFE-based optimal encodings, we again observed consistent performance during CV (Figures 5A-J) and independent validation (Figure 5K-T). Specifically, the top three models on training and IVDS were based on optimal hybrid, QSO, and PseAAC encodings, and their ACC scores on IVDS are 0.888, 0.863, and 0.856, respectively. Among these, the model using hybrid encodings was based on XGBOOST, whereas the other two encodings utilized an SVM classifier. Although both feature selection techniques displayed consistent performance during CV and independent validation, the performance achieved by the SVM model using Boruta-based optimal hybrid encodings was 3–5% better than the XGBOOST model using RFE-based optimal hybrid encodings. Therefore, the SVM model based on optimal hybrid encodings derived from the Boruta algorithm was selected as the final model.

F. Comparison of Boruta-based optimal encodings with control and excluded features on the training dataset

Next, to feature dimension reduction, we attempted to assess if the optimal encodings exhibit better performance than control (all features) or excluded descriptors for each feature, we developed several classification models based on control and excluded features using the training dataset. The comparative analysis demonstrates a minor improvement in the performance of most encodings, and a significant increase in the performance of KNN and SVM-based classifiers when optimal hybrid encodings were used to build the models (Figure S1). Specifically, a 30–40% increase compared to control and excluded features was observed in these two models on Boruta-based optimal encodings. Similarly, about a 16% increase in performance was observed when optimal DPC-based encodings were tested using an SVM classifier. Interestingly, there was a 1–2% decrease in the performance of a few models when optimal feature sets were used. These include XGBOOST-based models using AutoC, DPC, PseAAC, and QSO encodings. Similarly, a 1% decrease was observed using CTD encodings when an RF-based classifier was used to generate the model. In contrast, the performance of all the models decreased substantially when models were generated using excluded features (Figure S1). These data suggest that the Boruta algorithm identified relevant encodings that improve performance and dimension reduction.

G. Model availability

At present, there is no freely accessible tool or standalone software leveraging sequence-based optimal features for predicting TP proteins. This limitation hinders the application of innovative methods in effectively complementing the experimental characterization and annotation of these sequences. To address the disparity between sequencing and the functional annotation of potential TP proteins, we have developed a user-friendly web server that facilitates the identification of these proteins. This tool is openly available to users at https://procarb.org/TP-ML/.

IV. Discussion

Thirty years ago, Muggleton et al. [73] exploited an ML-based approach to predict the secondary structure of the protein. Since then, ML methodologies have undergone significant advancements in their capacity to learn from intricate datasets and construct diverse prediction models [74]. Over this period, the utilization of ML in protein science has witnessed a significant increase in addressing the problems related to protein structure prediction [75], protein classification [33], [76], [77],

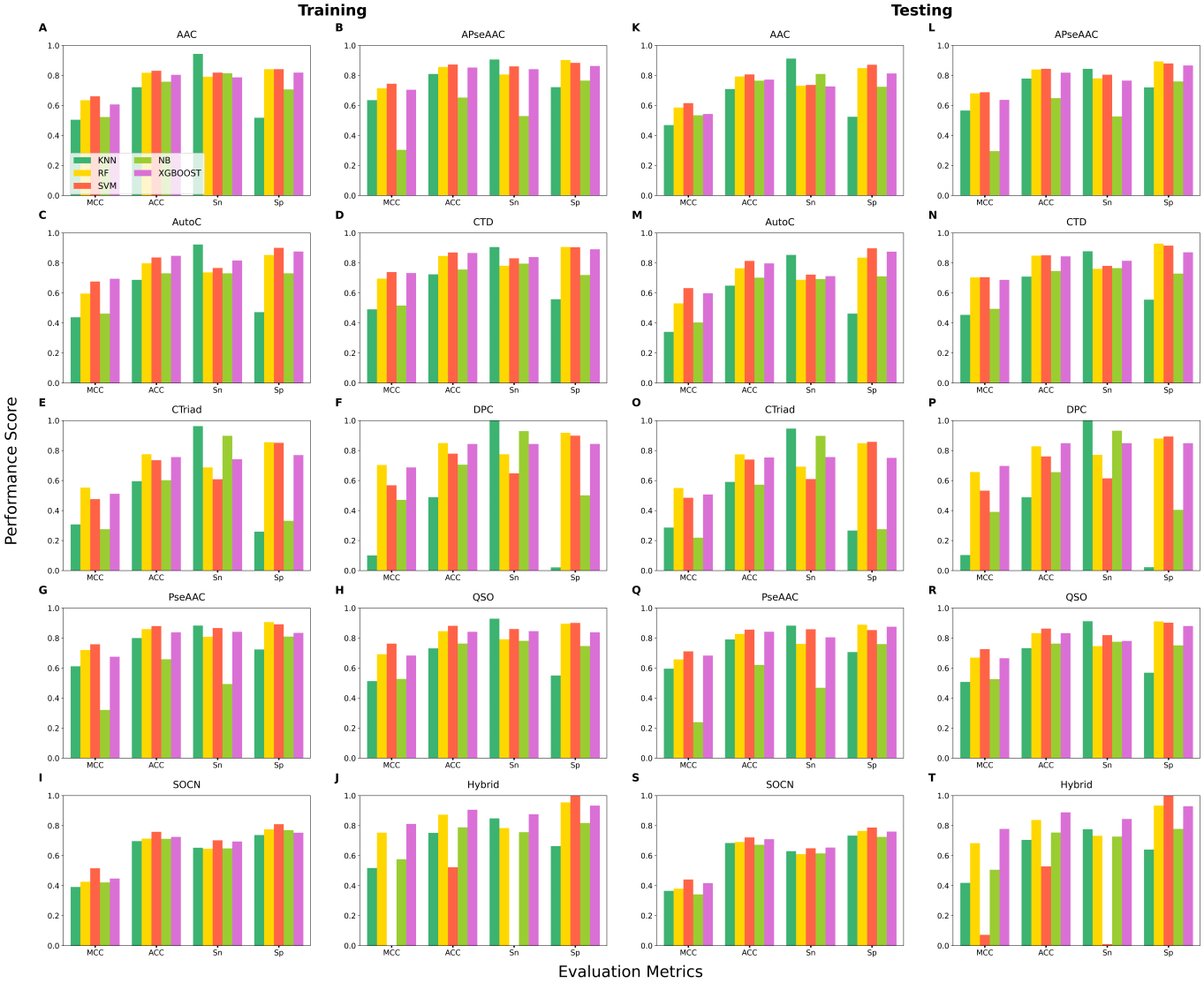


Fig. 5: Performance comparison of five ML classifiers and RFE-based optimal feature encodings on CV (panels A to J) and independent validation (panels K to T) for AAC, APseAAC, AutoC, CTD, CTriad, DPC, PseAAC, QSO, SOCN, and Hybrid feature descriptors, respectively.

peptide therapeutics [37], [78], [79], or prediction of binding sites [80], [81]. In light of this, ML applications have been successfully developed for predicting different proteases such as sortases [32], the C10 family [34], and asparagine peptide lyases [35]. Considering the importance of TPs in regulating cellular function and the absence of existing ML-based models for predicting TPs from their primary AA sequences, we have constructed an SVM-based predictor, named TP-ML, to distinguish TPs from non-TPs utilizing sequence-derived optimal features.

To construct TP-ML, we systematically generated and assessed the performance of different classifiers using a non-redundant balanced dataset. We selected the model based on its consistent performance across CV and independent validation experiments. Given that high-dimensional feature sets often contain irrelevant or superfluous elements, potentially affecting classifier performance [82], we employed two distinct feature selection techniques: (i) Boruta and (ii) RFE, to select the optimal feature subset for each descriptor. Subsequently, we input these subsets into five different ML classifiers and compared their performance. This rigorous comparative analysis enabled us to identify and adopt the most appropriate ML classifier equipped with optimal features.

TP-ML represents the only freely available predictor to predict TP proteins. While it demonstrates satisfactory performance, future advancements could involve improving the model's robustness such as constructing models using larger datasets, exploring alternative feature encodings, and developing ensemble-based models [83]. Furthermore, TP-ML cannot currently classify distinct members of TP families (e.g., T1, T2, T3, etc.), primarily due to the current scarcity of sequences representing specific TP types in public databases. Therefore, with the availability of more data in the future, we intend to develop a two-layer hybrid model [32]–[34]. The first layer

will predict whether a given sequence is a TP protein, and the second layer will leverage the predicted TP sequences to categorize them into specific TP family members.

V. Conclusion

TPs are a group of proteolytic enzymes characterized by the presence of a threonine residue situated in their active sites. These enzymes serve as the central catalytic components within the proteasome, a huge protein-degrading machinery. Their critical roles are recognized in diseases including Alzheimer's and cancer. However, the identification and characterization of TPs face significant experimental challenges. To address this issue, we have proposed TP-ML to effectively classify TP from non-TP sequences using optimal hybrid encodings extracted from their primary AA sequences and SVM classifier. The proposed TP-ML method demonstrated stable performance during both training and independent evaluation. It is worth noting that TP-ML is the first method to predict TP proteins. To facilitate usage, we have made it publicly available as a web server at https://procarb.org/TP-ML/, providing the community with a tool to identify potential putative TPs from experimental results.

References

[1] A. Eatemadi, H. T. Aiyelabegan, B. Negahdari, M. A. Mazlomi, H. Daraee, N. Daraee, R. Eatemadi, and E. Sadroddiny, "Role of protease and protease inhibitors in cancer pathogenesis and treatment," *Biomedicine & Pharmacotherapy*, vol. 86, pp. 221–231, 2017.

[2] Y. Yang, H. Hong, Y. Zhang, and W. Cai, "Molecular imaging of proteases in cancer," *Cancer growth and metastasis*, vol. 2, pp. CGM–S2814, 2009.

[3] N. D. Rawlings, A. J. Barrett, and A. Bateman, "Using the merops database for proteolytic enzymes and their inhibitors and substrates," *Current protocols in bioinformatics*, vol. 48, no. 1, pp. 1–25, 2014.

[4] B. Turk, "Targeting proteases: successes, failures and future prospects," *Nature Reviews Drug Discovery*, vol. 5, no. 9, pp. 785–799, 2006.

[5] P. Philipps-Wiemann, "Proteases—general aspects," in *Enzymes in human and animal nutrition*. Elsevier, 2018, pp. 257–266.

[6] C. Veltri, "Proteases: nature's destroyers and the drugs that stop them," *Pharm. Pharmacol. Int. J*, vol. 2, no. 6, pp. 1–11, 2015.

[7] C. S. Craik, M. J. Page, and E. L. Madison, "Proteases as therapeutics," *Biochemical Journal*, vol. 435, no. 1, pp. 1–16, 2011.

[8] S. Rakashanda, F. Rana, S. Rafiq, A. Masood, and S. Amin, "Role of proteases in cancer: A review," *Biotechnol Mol Biol Rev*, vol. 7, no. 4, pp. 90–101, 2012.

[9] M. Bochtler, L. Ditzel, M. Groll, C. Hartmann, and R. Huber, "The proteasome," *Annual review of biophysics and biomolecular structure*, vol. 28, no. 1, pp. 295–317, 1999.

[10] N. D. Rawlings, A. J. Barrett, P. D. Thomas, X. Huang, A. Bateman, and R. D. Finn, "The merops database of proteolytic enzymes, their substrates and inhibitors in 2017 and a comparison with peptidases in the panther database," *Nucleic acids research*, vol. 46, no. D1, pp. D624–D632, 2018.

[11] L. D. Fricker, "Proteasome inhibitor drugs," *Annual review of pharmacology and toxicology*, vol. 60, no. 1, pp. 457–476, 2020.

[12] J. C. Powers, J. L. Asgian, Ö. D. Ekici, and K. E. James, "Irreversible inhibitors of serine, cysteine, and threonine proteases," *Chemical reviews*, vol. 102, no. 12, pp. 4639–4750, 2002.

[13] A. J. McClellan, S. H. Laugesen, and L. Ellgaard, "Cellular functions and molecular mechanisms of non-lysine ubiquitination," *Open Biology*, vol. 9, no. 9, p. 190147, 2019.

[14] L. Budenholzer, C. L. Cheng, Y. Li, and M. Hochstrasser, "Proteasome structure and assembly," *Journal of molecular biology*, vol. 429, no. 22, pp. 3500–3524, 2017.

[15] Y.-Y. Li, Y.-L. Bao, Z.-B. Song, L.-G. Sun, P. Wu, Y. Zhang, C. Fan, Y.-X. Huang, Y. Wu, C.-L. Yu *et al.*, "The threonine protease activity of testes-specific protease 50 (tsp50) is essential for its function in cell proliferation," *PloS one*, vol. 7, no. 5, p. e35030, 2012.

[16] J. Shan, L. Yuan, Q. Xiao, N. Chiorazzi, D. Budman, S. Teichberg, and H.-p. Xu, "Tsp50, a possible protease in human testes, is activated in breast cancer epithelial cells," *Cancer research*, vol. 62, no. 1, pp. 290–294, 2002.

[17] H.-P. Xu, L. Yuan, J. Shan, and H. Feng, "Localization and expression of tsp50 protein in human and rodent testes," *Urology*, vol. 64, no. 4, pp. 826–832, 2004.

[18] L. Zheng, G. Xie, G. Duan, X. Yan, and Q. Li, "High expression of testes-specific protease 50 is associated with poor prognosis in colorectal carcinoma," *PloS one*, vol. 6, no. 7, p. e22203, 2011.

[19] H. Xu, J. Shan, V. Jurukovski, L. Yuan, J. Li, and K. Tian, "Tsp50 encodes a testis-specific protease and is negatively regulated by p53," *Cancer research*, vol. 67, no. 3, pp. 1239–1245, 2007.

[20] J. Ke, J. Lou, R. Zhong, X. Chen, J. Li, C. Liu, Y. Gong, Y. Yang, Y. Zhu, Y. Zhang *et al.*, "Identification of a potential regulatory variant for colorectal cancer risk mapping to 3p21. 31 in chinese population," *Scientific Reports*, vol. 6, no. 1, p. 25194, 2016.

[21] O. Ward, "Proteases," *Comprehensive biotechnology*, p. 571, 2011.

[22] R. P. Dyer and G. A. Weiss, "Making the cut with protease engineering," *Cell chemical biology*, vol. 29, no. 2, pp. 177–190, 2022.

[23] L. Steward, M. F. Brin, and A. Brideau-Andersen, "Novel native and engineered botulinum neurotoxins," *Botulinum Toxin Therapy*, pp. 63–89, 2020.

[24] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.

[25] N. Xiao, D.-S. Cao, M.-F. Zhu, and Q.-S. Xu, "protr/protrweb: R package and web server for generating various numerical representation schemes of protein sequences," *Bioinformatics*, vol. 31, no. 11, pp. 1857–1859, 2015.

[26] M. M. Gromiha, *Protein bioinformatics: from sequence to function*. academic press, 2011.

[27] K.-C. Chou, "Prediction of protein cellular attributes using pseudo-amino acid composition," *Proteins: Structure, Function, and Bioinformatics*, vol. 43, no. 3, pp. 246–255, 2001.

[28] K. C. Chou, "Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes," *Bioinformatics*, vol. 21, no. 1, pp. 10–19, 2005.

[29] C. Chen, Q. Zhang, Q. Ma, and B. Yu, "Lightgbm-ppi: Predicting protein-protein interactions through lightgbm with multi-information fusion," *Chemometrics and Intelligent Laboratory Systems*, vol. 191, pp. 54–64, 2019.

[30] I. Dubchak, I. Muchnik, C. Mayor, I. Dralyuk, and S.-H. Kim, "Recognition of a protein fold in the context of the scop classification," *Proteins: structure, function, and bioinformatics*, vol. 35, no. 4, pp. 401–407, 1999.

[31] I. Dubchak, I. Muchnik, S. R. Holbrook, and S.-H. Kim, "Prediction of protein folding class using global description of amino acid sequence." *Proceedings of the National Academy of Sciences*, vol. 92, no. 19, pp. 8700–8704, 1995.

[32] A. Malik, S. Subramaniyam, C.-B. Kim, and B. Manavalan, "Sortpred: The first machine learning based predictor to identify bacterial sortases and their classes using sequence-derived information," *Computational and structural biotechnology journal*, vol. 20, pp. 165–174, 2022.

[33] A. Firoz, A. Malik, H. M. Ali, Y. Akhter, B. Manavalan, and C.-B. Kim, "Prr-hypred: A two-layer hybrid framework to predict pattern recognition receptors and their families by employing sequence encoded optimal features," *International Journal of Biological Macromolecules*, vol. 234, p. 123622, 2023.

[34] A. Malik, N. Mahajan, T. A. Dar, and C.-B. Kim, "C10pred: A first machine learning based tool to predict c10 family cysteine peptidases using sequence-derived features," *International Journal of Molecular Sciences*, vol. 23, no. 17, p. 9518, 2022.

[35] A. Malik, M. R. Kamli, J. S. Sabir, I. A. Rather, C.-B. Kim, B. Manavalan *et al.*, "Aplpred: A machine learning-based tool for accurate prediction and characterization of asparagine peptide lyases using sequence-derived optimal features," *Methods*, vol. 229, pp. 133–146, 2024.

[36] N. T. Pham, L. T. Phan, J. Seo, Y. Kim, M. Song, S. Lee, Y.-J. Jeon, and B. Manavalan, "Advancing the accuracy of sars-cov-2 phosphorylation site detection via meta-learning approach," *Briefings in Bioinformatics*, vol. 25, no. 1, p. bbad433, 2024.

[37] L. T. Phan, H. W. Park, T. Pitti, T. Madhavan, Y.-J. Jeon, B. Manavalan *et al.*, "Mlacp 2.0: An updated machine learning tool for anticancer peptide prediction," *Computational and Structural Biotechnology Journal*, vol. 20, pp. 4473–4480, 2022.

[38] N. Bupi, V. K. Sangaraju, L. T. Phan, A. Lal, T. T. B. Vo, P. T. Ho, M. A. Qureshi, M. Tabassum, S. Lee, and B. Manavalan, "An effective

integrated machine learning framework for identifying severity of tomato yellow leaf curl virus and their experimental validation," *Research*, vol. 6, p. 0016, 2023.

[39] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, and H. Jiang, "Predicting protein-protein interactions based only on sequences information," *Proceedings of the National Academy of Sciences*, vol. 104, no. 11, pp. 4337–4341, 2007.

[40] X.-Y. Jing and F.-M. Li, "Predicting cell wall lytic enzymes using combined features," *Frontiers in Bioengineering and Biotechnology*, vol. 8, p. 627335, 2021.

[41] K.-C. Chou, "Prediction of protein subcellular locations by incorporating quasi-sequence-order effect," *Biochemical and biophysical research communications*, vol. 278, no. 2, pp. 477–483, 2000.

[42] J. Dong, M.-F. Zhu, Y.-H. Yun, A.-P. Lu, T.-J. Hou, and D.-S. Cao, "Biomedr: an r/cran package for integrated data analysis pipeline in biomedical study," *Briefings in bioinformatics*, vol. 22, no. 1, pp. 474–484, 2021.

[43] S. Akbar, A. U. Rahman, M. Hayat, and M. Sohail, "cacp: Classifying anticancer peptides using discriminative intelligent model via chou's 5-step rules and general pseudo components," *Chemometrics and Intelligent Laboratory Systems*, vol. 196, p. 103912, 2020.

[44] S. A. Ong, H. H. Lin, Y. Z. Chen, Z. R. Li, and Z. Cao, "Efficacy of different protein descriptors in predicting protein functional families," *Bmc Bioinformatics*, vol. 8, pp. 1–14, 2007.

[45] B. A. van den Berg, M. J. Reinders, J. A. Roubos, and D. d. Ridder, "Spice: a web-based tool for sequence-based protein classification and exploration," *BMC bioinformatics*, vol. 15, pp. 1–10, 2014.

[46] M. Kuhn, "Building predictive models in r using the caret package," *Journal of statistical software*, vol. 28, pp. 1–26, 2008.

[47] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[48] H. Shen and K.-C. Chou, "Using optimized evidence-theoretic k-nearest neighbor classifier and pseudo-amino acid composition to predict membrane protein types," *Biochemical and biophysical research communications*, vol. 334, no. 1, pp. 288–292, 2005.

[49] K. Syed, W. C. Sleeman IV, J. J. Nalluri, R. Kapoor, M. Hagan, J. Palta, and P. Ghosh, "Artificial intelligence methods in computer-aided diagnostic tools and decision support analytics for clinical informatics," in *Artificial Intelligence in Precision Health*. Elsevier, 2020, pp. 31–59.

[50] D. Saha and A. Manickavasagan, "Machine learning techniques for analysis of hyperspectral images to determine quality of food products: A review," *Current Research in Food Science*, vol. 4, pp. 28–44, 2021.

[51] D. Saini, T. Chand, D. K. Chouhan, and M. Prakash, "A comparative analysis of automatic classification and grading methods for knee osteoarthritis focussing on x-ray images," *Biocybernetics and Biomedical Engineering*, vol. 41, no. 2, pp. 419–444, 2021.

[52] D. Rice, "Causal reasoning," *Calculus of Thought, Elsevier*, pp. 95–123, 2014.

[53] Z. Abbas, H. Tayara, and K. T. Chong, "Alzheimer's disease prediction based on continuous feature representation using multi-omics data integration," *Chemometrics and Intelligent Laboratory Systems*, vol. 223, p. 104536, 2022.

[54] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[55] T. Jo and J. Cheng, "Improving protein fold recognition by random forest," in *BMC bioinformatics*, vol. 15. Springer, 2014, pp. 1–7.

[56] J. Li, J. Wu, K. Chen *et al.*, "Pfp-rfsm: protein fold prediction by using random forests and sequence motifs," *Journal of Biomedical Science and Engineering*, vol. 6, no. 12, p. 1161, 2013.

[57] M. Waris, K. Ahmad, M. Kabir, and M. Hayat, "Identification of dna binding proteins using evolutionary profiles position specific scoring matrix," *Neurocomputing*, vol. 199, pp. 154–162, 2016.

[58] X. Ma, J. Guo, and X. Sun, "Dnabp: Identification of dna-binding proteins based on feature selection using a random forest and predicting binding residues," *PloS one*, vol. 11, no. 12, p. e0167345, 2016.

[59] M. Hayat, A. Khan, and M. Yeasin, "Prediction of membrane proteins using split amino acid and ensemble classification," *Amino acids*, vol. 42, pp. 2447–2460, 2012.

[60] M. F. Sabooh, N. Iqbal, M. Khan, M. Khan, and H. Maqbool, "Identifying 5-methylcytosine sites in rna sequence using composite encoding feature into chou's pseknc," *Journal of theoretical biology*, vol. 452, pp. 1–9, 2018.

[61] V. N. Vapnik, V. Vapnik *et al.*, "Statistical learning theory," 1998.

[62] J. C. Tong and S. Ranganathan, *Computer-aided vaccine design*. Elsevier, 2013.

[63] S. Ahmed, M. Arif, M. Kabir, K. Khan, and Y. D. Khan, "Predaodp: accurate identification of antioxidant proteins by fusing different descriptors based on evolutionary information with support vector machine," *Chemometrics and Intelligent Laboratory Systems*, vol. 228, p. 104623, 2022.

[64] S. Akbar and M. Hayat, "imethyl-sttnc: Identification of n6-methyladenosine sites by extending the idea of saac into chou's pseaac to formulate rna sequences," *Journal of theoretical biology*, vol. 455, pp. 205–211, 2018.

[65] F. Ali, M. Arif, Z. U. Khan, M. Kabir, S. Ahmed, and D.-J. Yu, "Sdbp-pred: Prediction of single-stranded and double-stranded dna-binding proteins by extending consensus sequence and k-segmentation strategies into pssm," *Analytical biochemistry*, vol. 589, p. 113494, 2020.

[66] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[67] A. Banjar, F. Ali, O. Alghushairy, and A. Daud, "idbp-pbmd: A machine learning model for detection of dna-binding proteins by extending compression techniques into evolutionary profile," *Chemometrics and Intelligent Laboratory Systems*, vol. 231, p. 104697, 2022.

[68] N. Q. K. Le, D. T. Do, Q. A. Le *et al.*, "A sequence-based prediction of kruppel-like factors proteins using xgboost and optimized features," *Gene*, vol. 787, p. 145643, 2021.

[69] H. Jeon and S. Oh, "Hybrid-recursive feature elimination for efficient feature selection," *Applied Sciences*, vol. 10, no. 9, p. 3211, 2020.

[70] M. B. Kursa and W. R. Rudnicki, "Feature selection with the boruta package," *Journal of statistical software*, vol. 36, pp. 1–13, 2010.

[71] A. Acharjee, J. Larkman, Y. Xu, V. R. Cardoso, and G. V. Gkoutos, "A random forest based biomarker discovery and power analysis framework for diagnostics research," *BMC medical genomics*, vol. 13, pp. 1–14, 2020.

[72] C. P. Cantalapiedra, A. Hernández-Plaza, I. Letunic, P. Bork, and J. Huerta-Cepas, "eggnog-mapper v2: functional annotation, orthology assignments, and domain prediction at the metagenomic scale," *Molecular biology and evolution*, vol. 38, no. 12, pp. 5825–5829, 2021.

[73] S. Muggleton, R. King, and M. Sternberg, "Protein secondary structure prediction using logic-based machine learning," *Protein Engineering, Design and Selection*, vol. 6, no. 5, pp. 549–549, 1993.

[74] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, "Definitions, methods, and applications in interpretable machine learning," *Proceedings of the National Academy of Sciences*, vol. 116, no. 44, pp. 22 071–22 080, 2019.

[75] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with alphafold," *nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[76] Y.-H. Wang, Y.-F. Zhang, Y. Zhang, Z.-F. Gu, Z.-Y. Zhang, H. Lin, and K.-J. Deng, "Identification of adaptor proteins using the anova feature selection technique," *Methods*, vol. 208, pp. 42–47, 2022.

[77] F.-Y. Dao, M.-L. Liu, W. Su, H. Lv, Z.-Y. Zhang, H. Lin, and L. Liu, "Acrpred: A hybrid optimization with enumerated machine learning algorithm to predict anti-crispr proteins," *International journal of biological macromolecules*, vol. 228, pp. 706–714, 2023.

[78] B. Manavalan and M. C. Patra, "Mlcpp 2.0: an updated cell-penetrating peptides and their uptake efficiency predictor," *Journal of Molecular Biology*, vol. 434, no. 11, p. 167604, 2022.

[79] H. Kurata, S. Tsukiyama, and B. Manavalan, "iacvp: markedly enhanced identification of anti-coronavirus peptides using a dataset-specific word2vec model," *Briefings in bioinformatics*, vol. 23, no. 4, p. bbac265, 2022.

[80] A. Firoz, A. Malik, K. H. Joplin, Z. Ahmad, V. Jha, and S. Ahmad, "Residue propensities, discrimination and binding site prediction of adenine and guanine phosphates," *BMC biochemistry*, vol. 12, pp. 1–12, 2011.

[81] A. Malik and S. Ahmad, "Sequence and structural features of carbohydrate binding in proteins and assessment of predictability using a neural network," *BMC Structural Biology*, vol. 7, pp. 1–14, 2007.

[82] S. Basith, G. Lee, and B. Manavalan, "Stallion: a stacking-based ensemble learning framework for prokaryotic lysine acetylation site prediction," *Briefings in Bioinformatics*, vol. 23, no. 1, p. bbab376, 2022.

[83] W.-R. Qiu, A. Xu, Z.-C. Xu, C.-H. Zhang, and X. Xiao, "Identifying acetylation protein by fusing its pseaac and functional domain annotation," *Frontiers in Bioengineering and Biotechnology*, vol. 7, p. 311, 2019.

Ahmad Firoz received an MSc degree from Jamia Milla Islamia University, India, and a PhD degree from Thapar University, India, in 2016. He is currently an assistant professor with the Department of Biological Sciences, King Abdulaziz University, Jeddah, Saudi Arabia. Before that, he worked as a scientist at the Post Graduate Institute of Medical Education and Research, Chandigarh, India. His research interests are molecular docking and simulations, development of bioinformatics databases and tools, machine-learning, and scientific computing.

Adeel Malik received the MSc and PhD degrees from Jamia Milla Islamia University, New Delhi, India, in 2004 and 2009, respectively. He is currently working as a research professor at the Institute of Intelligence Informatics Technology, Sangmyung University, Seoul, South Korea. His research interests include machine-learning, computational biology, and comparative genomics.

Nitin Mahajan received an MSc (Microbiology) degree from Central Research Institute, Kasauli, India, in 2002, an MS in Data Science from Bellevue University, USA, in 2020, and a Ph.D. degree in Experimental Medicine and Biotechnology from the Post-graduate Institute of Medical Education and Research (PGIMER), Chandigarh, India, in 2009. Since 2020, Nitin has worked as a Principal Data Scientist at Wugen Inc., USA, an off-the-shelf cell therapy company. Before that, he held a faculty position at Washington Univesity (WashU) in St Louis, USA. He also worked as a post-doctoral researcher at Northwestern University, Chicago. His research interests include cell therapy, next-generation sequencing, precision medicine, predictive modeling, and machine-learning.

Le Thi Phan received her bachelor's from Ha Noi University of Pharmacy (Vietnam) and an MS in Biotechnology from Chonnam National University (South Korea). She is currently a Ph.D. student in the Computational Biology and Bioinformatics Laboratory, Department of Integrative Biotechnology, Sungkyunkwan University (South Korea). Her research focuses on computational biology, bioinformatics, and data analytics in genome sequencing, drug discovery, and immunity.

Hani S.H. Mohammed Ali received an MS degree from King Abdulaziz University, Jeddah Saudi Arabia, and a PhD degree in Computational Biochemistry and Bioinformatics from the University of Essex, UK. Currently, he is working as an associate professor in the Department of Biological Sciences, at King Abdulaziz University, Jeddah, Saudi Arabia.

Chang-Bae Kim received a PhD degree in Zoology from Seoul National University (SNU) in 1991. He is now a professor at Sangmyung University. He was a senior researcher with the Korea Research Institute of Bioscience and Biotechnology (KRIBB). He also worked as a postdoctoral researcher at Yale University. Research interests of his research group include multi-omics data and environmental DNA analyses based on machine-learning methods.

Balachandran Manavalan received a PhD degree in Computational Biology from Ajou University in 2011. He is an Assistant Professor at the Department of Integrative Biotechnology, Sungkyunkwan University, Republic of Korea. He is also an associate member of the Korea Institute for Advanced Study, Republic of Korea. His research interests include artificial intelligence, bioinformatics, machine-learning, big data, and functional genomics.