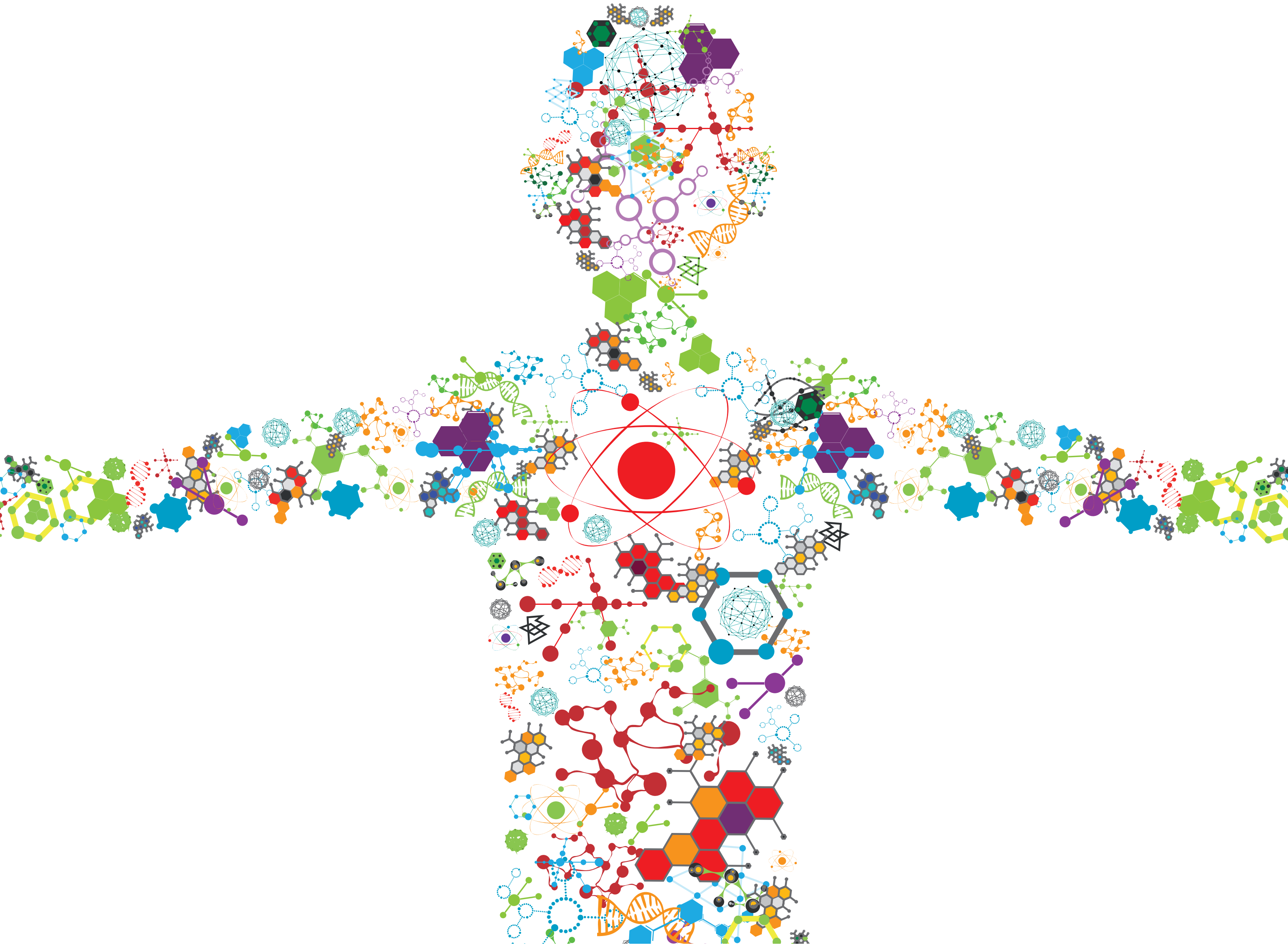
FEATURE REPRESENTATION AND LEARNING METHODS WITH APPLICATIONS IN PROTEIN SECONDARY STRUCTURE

EDITED BY: Zhibin Lv, Hong Wenjing and Xue Xu
PUBLISHED IN: Frontiers in Bioengineering and Biotechnology



frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-88971-555-8
DOI 10.3389/978-2-88971-555-8

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: frontiersin.org/about/contact

FEATURE REPRESENTATION AND LEARNING METHODS WITH APPLICATIONS IN PROTEIN SECONDARY STRUCTURE

Topic Editors:
Zhibin Lv, Sichuan University, China
Hong Wenjing, Xiamen University, China
Xue Xu, Wuhan University of Science and Technology, China

Citation: Lv, Z., Wenjing, H., Xu, X., eds. (2021). Feature Representation and Learning Methods With Applications in Protein Secondary Structure. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88971-555-8

Table of Contents

frontiers in Bioengineering and Biotechnology

EDITORIAL
published: 09 September 2021
doi: 10.3389/fbioe.2021.748722

Editorial: Feature Representation and Learning Methods With Applications in Protein Secondary Structure

Ni Yan[1], Zhibin Lv[2]*, Wenjing Hong[3]* and Xue Xu[1]*

[1]School of Medicine, Wuhan University of Science and Technology, Wuhan, China, [2]Department of Medical Instruments and Information, College of Biomedical Engineering, Sichuan University, Chengdu, China, [3]State Key Laboratory of Physical Chemistry of Solid Surfaces, iChEM, Xiamen University, Xiamen, China

Keywords: protein secondary structure, structure prediction, deep learning, machine learning, feature extraction

Editorial on the Research Topic

Feature Representation and Learning Methods With Applications in Protein Secondary Structure

In recent years, the rise of machine learning methods, especially deep learning, had greatly promoted the development of prediction of protein secondary structures. Such methods could not only make better use of exponentially growing massive protein sequence data, but were also able to automatically mine complex and latent patterns hidden in the data. Although significant progress had been made, we still faced challenges how to predict protein secondary structures directly from protein sequences with improved accuracy.

There were 11 articles published in the special issue *Feature Representation and Learning Methods With Applications in Protein Secondary Structure*. The authors here described computer methods and techniques for protein secondary structure predictions. Also, they presented and discussed latest algorithms development in feature extraction, dimension reduction, unbalanced classification, etc. The papers provided good references to those new to the field as well as experienced researchers.

Guo et al. established a model to classify thermophilic proteins and non-thermophilic proteins based on sequences. After feature extraction by iFeature, MRMD2.0 was applied for feature selection and dimension reduction, and LIBSVM was used to obtain the optimal parameters of the model and established the prediction model. Compared with LMT, Logistic, Random Forest, BayesNet, REPTree, J48, the prediction rate of this model was the highest (SE: 95.85%, SP: 96.22%, ACC: 96.02%).

Li et al. constructed a model to identify antioxidant proteins based on a support vector machine based method, Vote9. Sequence features were extracted by using reduced amino acid compositions and the optimal g-gap dipeptide compositions from nine optimal individual models.

Gu et al. distinguished GPCRs and non-GPCRs with CTDC extraction and MRMD2. 0 dimension-reduction. The authors found different methods of feature extraction and the same method of dimensionality reduction had different effects on distinguishing GPCRs and non-GPCRs. The correct classification rate of five independent test sets was 90.64, 90.37, 88.04, 93.28, and 95.73%, with an average rate of 91.61 ± 2.96%.

Jing and Li used amino acid composition, dipeptide composition, position-specific score matrix auto-covariance, and Auto-covariance average chemical shift to predict cell wall lytic enzymes. SMOTE was used to counter the imbalanced data classification problems, and F-score algorithm was used to remove redundant or irrelevant features. ACC was 99.19% with jackknife test.

Edited and reviewed by:
Jean Marie François,
Institut Biotechnologique de Toulouse (INSA), France

***Correspondence:**
Zhibin Lv
lvzhibin@pku.edu.cn
Wenjing Hong
whong@xmu.edu.cn
Xue Xu
xuexu007@wust.edu.cn

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: 28 July 2021
Accepted: 27 August 2021
Published: 09 September 2021

Citation:
Yan N, Lv Z, Hong W and Xu X (2021) Editorial: Feature Representation and Learning Methods With Applications in Protein Secondary Structure. Front. Bioeng. Biotechnol. 9:748722. doi: 10.3389/fbioe.2021.748722

Chen et al. proposed a novel computational model for lncRNA-protein interaction relationship prediction based on machine learning methods. A method for representing the topological feature information of the network of lncRNA-protein interaction was proposed. Protein evolutionary information, protein CTD sequence information features, lncRNA sequence mutual information features, and lncRNA expression profile information were extracted, and the recursive feature elimination algorithm was used to optimize feature vectors. The obtained optimized feature vectors were fed into SVM to predict lncRNA-protein interactions. This method was experimentally compared with six excellent lncRNA-protein prediction algorithms, and experimental results showed that our proposed method achieves the best performance values in AUPR (74.39%) and F1 score (65.91%).

Li et al. used a total of 12 feature extraction methods when predicted anticancer peptides. After eight times of dimension reduction by MRMD2.0, they established a 19-dimensional feature model based on anticancer peptide sequences, which had lower dimension and better performance (ACC: 92.15–92.73%, SE: 85.5–87.7%, SP: 96.1–97.1%, MCC: 83.7–84.9%, F1 score: 92.1–92.7%) than some existing methods.

Wang et al. developed a bioinformatics tool called prPred for the prediction of plant resistance proteins that combines CKSAAP and CKSAAGP features based on SVM. Experimental results showed that the accuracy, precision, sensitivity, specificity, F1-score, MCC, and AUC of prPred were 0.935, 1.000, 0.806, 1.000, 0.893, 0.857, and 0.948, respectively, on an independent test set. The predictive and analytical results demonstrated that the constructed model was an efficient predictor to distinguish R proteins from non-R proteins.

Cai et al. established a comprehensive weight model SDN2GO based on protein sequence, protein domain content and known protein-protein interaction network. Compared with NetGO, DeepGO and the classic BLAST method, the authors' results showed that SDN2GO achieved the maximum F-max value (36.1–56.1%) of each sub ontology of GO.

Liu et al. established a deep learning-based predictor TMPSS to predict the secondary structure and topological structure of α-helical TMPs. The TMPSS applied a deep learning network that included grouped multi-scale CNN (Convolutional Neural Network) and stacked attention-enhanced BiLSTM (Bidirectional Long Short-Term Memory) layers to capture local and global context. Based on the multi-task learning method, the prediction performance was improved and the amount of calculation was reduced by considering the interaction between different protein properties.

Yallapragada et al. established a game-based molecular visualization tool PePblock Builder VR-AN. Different from traditional sequence-based protein designs and fragment-based splicing, pepblockbuilder-VR provided a building block environment for the construction of complex structures, which provided users with a unique visual structure construction experience. In addition, Pepblock Builder VR worked as an independent and VR-based application and provided us with a good platform for teaching.

Lyu et al. established a reductive deep learning model MLPRNN to predict either 3-state or 8-state protein secondary structures, which had the same prediction accuracy as DeepCNF, MUFOLD-SS, BGRUCB, CRRNN and DNSS2.

The 11 papers in this research topic covered only a small part of the computer methods and techniques used to predict protein secondary structure. We hope more and more researchers will devote their time and effort into this field to predict the secondary structure of proteins more quickly, simply and accurately.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

FUNDING

This work is supported by the National Natural Science Foundation of China (Grant nos. 62001090 and 82104695) and the Fundamental Research Funds for the Central Universities (no. YJ2021104).

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

SDN2GO: An Integrated Deep Learning Model for Protein Function Prediction

Yideng Cai[1], Jiacheng Wang[1] and Lei Deng[1,2*]

[1] School of Computer Science and Engineering, Central South University, Changsha, China, [2] School of Software, Xinjiang University, Urumqi, China

The assignment of function to proteins at a large scale is essential for understanding the molecular mechanism of life. However, only a very small percentage of the more than 179 million proteins in UniProtKB have Gene Ontology (GO) annotations supported by experimental evidence. In this paper, we proposed an integrated deep-learning-based classification model, named SDN2GO, to predict protein functions. SDN2GO applies convolutional neural networks to learn and extract features from sequences, protein domains, and known PPI networks, and then utilizes a weight classifier to integrate these features and achieve accurate predictions of GO terms. We constructed the training set and the independent test set according to the time-delayed principle of the Critical Assessment of Function Annotation (CAFA) and compared it with two highly competitive methods and the classic BLAST method on the independent test set. The results show that our method outperforms others on each sub-ontology of GO. We also investigated the performance of using protein domain information. We learned from the Natural Language Processing (NLP) to process domain information and pre-trained a deep learning sub-model to extract the comprehensive features of domains. The experimental results demonstrate that the domain features we obtained are much improved the performance of our model. Our deep learning models together with the data pre-processing scripts are publicly available as an open source software at https://github.com/Charrick/SDN2GO.

Keywords: protein function, word embedding, convolutional neural network, deep multi-label classification, deep learning

OPEN ACCESS

Edited by:
Zhibin Lv,
University of Electronic Science and Technology of China, China

Reviewed by:
Qingting Wei,
Nanchang University, China
Yanglan Gan,
Donghua University, China

****Correspondence:***
Lei Deng
leideng@csu.edu.cn

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *13 March 2020*
Accepted: *07 April 2020*
Published: *29 April 2020*

Citation:
Cai Y, Wang J and Deng L (2020) SDN2GO: An Integrated Deep Learning Model for Protein Function Prediction. Front. Bioeng. Biotechnol. 8:391. doi: 10.3389/fbioe.2020.00391

1. INTRODUCTION

As an essential structural molecule, protein is a vital component of all biological tissues and cells and is also the primary bearer of life activities (Weaver, 2011). Understanding protein function is important both for biology and medicine and pharmacy. For example, clarifying the function of a protein can provide a target for genetic manipulation, and provide a reliable basis for designing a new protein or transform an existing protein, etc. So that, accurate annotation of protein functions is a significant and crucial task. Traditional experimental methods require a lot of resources and time to determine protein function, despite there are high accuracy and reliability. With the continuous development of high-throughput sequencing technology and genomics, the sequence of proteins has been exploded, but just a small percentage of the total known and predicted protein sequences have been extensively annotated regarding their functions. Currently, only <0.1% of the more than 179 million proteins in UniProtKB have been experimentally annotated (Consortium, 2019). However, it isn't straightforward to scale up the experimental method to accommodate such a large amount of protein sequence data, which urgently requires the development of computational methods to assist to annotate protein functions (Radivojac et al., 2013).

Gene Ontology, launched in 1998, is widely used in the field of Bioinformatics, and the original intention of GO was to provide a representative platform for terminology description or interpretation of words of genes and gene product characteristics. It enables Bioinformatics researchers to summarize, process, interpret, and share the data of genes and gene products (Ashburner et al., 2000). Gene Ontology is a Directed Acyclic Graph (DAG) type ontology. At present, GO contains more than 45,000 biological concepts include functions and cell locations, and is divided into three categories, covering three aspects of biology: Biological Process, Molecular Function, and Cellular Component. A protein generally has multiple GO annotations; therefore, protein function prediction is a very large-scale multi-label classification problem (Zhang and Zhou, 2013), and accurately assigning GO terms to proteins is a challenging task.

In recent years, some organizations and teams have developed algorithms, tools, and systems for protein function prediction using advanced computer technologies, such as machine learning and deep neural networks (Kulmanov et al., 2018; You et al., 2018, 2019; Hakala et al., 2019; Lv et al., 2019b; Piovesan and Tosatto, 2019; Rifaioglu et al., 2019; Kulmanov and Hoehndorf, 2020). Researchers predict protein functions from one or more of the followings: protein sequences (Kulmanov et al., 2018; You et al., 2018, 2019; Hakala et al., 2019; Piovesan and Tosatto, 2019; Kulmanov and Hoehndorf, 2020), protein structures (Yang et al., 2015; Zhang et al., 2018), protein protein interactions (PPI) network (Kulmanov et al., 2018; Zhang et al., 2018; You et al., 2019), and others (Kahanda and Ben-Hur, 2017; Hakala et al., 2019; Piovesan and Tosatto, 2019; Rifaioglu et al., 2019). For example specifically, GOLabeler (You et al., 2018) integrated five different types of sequence-based information and learned from the idea of web page ranking to train an LTR (learning to rank) regression model to receive these five types of information to achieve accurate annotation of GO terms. As a result, this model got the best overall performance among all submissions of the 3rd Critical Assessment of Function Annotation (CAFA3). NetGO (You et al., 2019), proposed by the GOLabeler team, is based on GOLabeler and incorporates massive amounts of protein-protein interaction (PPI) network information into the LTR framework. Compared with GOLabler, it has achieved a significant improvement in protein function prediction performance. Hakala et al. (2019) developed an integrated system, which obtain features from several different tools or methods: BLASTP, InterproScan, NCBI Taxonomy, NucPred, NetAcet, PredGPI, and Amino Acid Index (Kawashima and Kanehisa, 2000; Heddad et al., 2004; Kiemer et al., 2005; Pierleoni et al., 2008; Camacho et al., 2009; Federhen, 2012; Jones et al., 2014), and then respectively feed all the features to two classifiers based on neural network and random forest and finally combined the NN classifier and the RF classifier to achieve the best prediction performance. DeepGO (Kulmanov et al., 2018) encodes the amino acid sequence of the protein by trigrams and maps the trigrams to vector by one-hot encoding and dense embedding, and then feed it to a convolutional neural network (CNN) to extract the feature map. Next, a combined feature vector consisting of CNN features and PPI Network embedding features entered into the hierarchically structured classification layers for classification of GO terms. INGA2.0 (Piovesan and Tosatto, 2019) uses four components, Homology which inferred from sequence similarity, Domain architecture, protein-protein interaction networks, and integrated information from the "dark proteome" which include disordered and transmembrane regions, to predict protein function. This method has better capabilities to predict some extremely rare GO terms compared with others. Overall, these highly competitive models and systems have proven their outstanding performance in protein function prediction and are continually being optimized.

The amino acid sequence is crucial for understanding and analyzing proteins of various species. Some studies have shown that sequence homology-based BLAST methods are highly competitive in protein function prediction (Altshul, 1997; Gillis and Pavlidis, 2013; Hamp et al., 2013). Besides, there are several high-level physiological functions, such as apoptosis or rhythm regulation, which are often the result of the interaction of multiple proteins (Kulmanov et al., 2018), and according to the so-called "guilt-by-association" principle, interacting proteins should have some similar functions (Oliver, 2000; Schwikowski et al., 2000). Those shows that protein sequence information and PPI network information are essential to predict protein function. We have also noticed the critical position of the protein domain in protein-related features. The domain is a structural motif that exists independently in different combinations, and orders in the protein (Forslund and Sonnhammer, 2008) and is a higher-level protein component than the amino acid sequence (Richardson, 1981). Therefore, it makes sense to analyze and examine the effect of Domain content on protein function and try to use it to predict protein function. Besides, Machine Learning (ML) is currently popular and efficient for bioinformatics problems (You et al., 2018, 2019; Lai et al., 2019; Tan et al., 2019; Wang et al., 2019a; Zhu et al., 2019; Dao et al., 2020), especially, due to its strong ability to fit high-dimensional, sparse, and highly collinear complex data, deep learning technology has been widely used in bioinformatics fields, such as protein structure and function (Sønderby and Winther, 2014; Spencer et al., 2014; Wei et al., 2018; Kulmanov and Hoehndorf, 2020), gene expression regulation (Chen et al., 2016; Lanchantin et al., 2016), protein classification (Asgari and Mofrad, 2015; Sønderby et al., 2015), and structure and functions of nucleic acid (Zhang et al., 2016; Lv et al., 2019a; Wang et al., 2019a,b). For these considerations, here we proposed an integrated deep learning model based on protein sequences, protein domain content, and known protein-protein interaction networks to predict protein function. We first built three different neural network modules to learn features from protein sequences, domain content, and PPI Net separately, and then combined the features from these three different sources and inputted them to the neural network classifier to predict the probability of each GO term. The experimental results show that our method of adding domain content to predict protein function is successful, and our model achieved better performance than BLAST and two other recent high-performance methods on an independent dataset constructed using time-delay rules.

2. MATERIALS AND METHODS

2.1. Data Source

2.1.1. Training Data

- Sequence Data

  For our experiments, we downloaded the sequence information of the proteins needed for the research from the UniProt database as FASTA-format files (http://www.uniprot.org/downloads) (Consortium, 2015). Then a CD-hit tool was used to de-redundant the downloaded protein sequence data. We grouped proteins with a sequence similarity >60% into one cluster, and only one protein per cluster was retained. Finally, we obtained a benchmark for humans contains 13,704 proteins, and a benchmark for Yeast contains 6,623 proteins.
- Annotation Data

  We downloaded GO annotation data for proteins from GOA (http://www.ebi.ac.uk/GOA) (Barrell et al., 2009) published in December 2013. Please note that the GO annotation data here is for training only, and all data are annotated in 2013 or earlier. Finally, the annotation data contains 13,882 categories (9,221 in BP, 3,483 in MF, and 1,178 in CC) for Human and 4,796 categories (2,439 in BP, 1,733 in MF, and 624 in CC) for Yeast.
- Protein-Protein interaction (PPI) Network Data

  We have added protein-protein interaction (PPI) network data, which is derived from the STRING database v10 (https://string-db.org/) (Szklarczyk et al., 2015), to improve the performance of the experiment. Among them, human PPI data contains 11,759,455 scored links of 19,257 proteins, and Yeast's PPI data contains 1,845,966 scored links of 6,507 proteins.
- Protein Domain Data

  We downloaded protein domain data from the public database interpro (Hunter et al., 2009) (http://www.ebi.ac.uk/interpro/download/), which contains the all UniProtKB proteins and the InterPro entries and individual signatures they match. For a specific protein, we can obtain the types, quantity, and locations of all the domains it contains, and the start and the end positions in the protein sequence of a domain are indicated. We searched by the protein's UniProt ID to obtain the domain data of all the proteins we needed. Next, we performed de-redundancy; for the same domain information supported by contradictory evidence, we kept only one of them. In the end, our domain data contains 113,972 pieces of information of 14,242 domains for Human, and 23,326 pieces of information of 6,707 domains for Yeast.

2.1.2. Independent Testing Data

The independent test data set is used for comparison with the competing methods. The collection of data generally follows the time-delayed rule of the CAFA challenge. We downloaded GO annotation data for proteins from GOA published in January 2016 and then obtained protein GO annotations added after 2013 (2014 and 2015). Specifically, we removed the annotation data published in December 2013 from the annotation data published in January 2016 and only retained the newly added protein annotation data. Next, we constructed an independent test benchmark based on the newly added annotation data; please note that all proteins contained in this benchmark do not have any GO annotations before 2014. Similarly, we filtered those proteins that were only annotated by GO terms that are extremely infrequent. The filtered independent test set contains 68 proteins for BP, 136 proteins for MF, and 106 proteins for CC.

2.2. Data Representation

2.2.1. Protein Sequence Data

Protein sequence information is one of the inputs to our model. The sequence of each protein is a string composed of 20 specific amino acid codes with different lengths. In this experiment, we only selected proteins with a sequence length not exceeding 1,500. If the sequence length is <1,500, we padded zero at the end of the sequence to ensure that the length of each input protein sequence information is fixed. To fully extract the context and semantic knowledge of the sequence, we utilized the ProtVec of BioVec (Asgari and Mofrad, 2015), which is a biological sequence representation and feature extraction method, to map the sequence information. This method borrows the ideas of "word embedding" from Natural Language Processing (NLP) and obtains vector representations of biological sequences through training, and ProtVec is used for protein sequences. We followed ProtVec and used 3-grams encoding for protein sequences, that is, using a window of length 3 with a step size of 1 to slide the protein sequence to obtain a 3-grams sequence with a length of 1498 for each protein.

In order to convert 3-grams sequences information into vectors that can be received by the computing model, we used the ProtVec-100d-3grams table released by BioVec. We Downloaded this data from Harvard Dataverse (http://dx.doi.org/10.7910/DVN/JMFHTN). In this table, the protein vector is a distributed representation of proteins, and a 100-D vector presents each 3-gram. For our experiment, according to ProtVec, each protein will be represented as a 1,498 * 100 vector matrix, and then used as input to the model. In particular, according to the way we treat proteins <1,500 in length, if a 3-gram word contains one or more zeros we have padded, then the 3-gram will be represented as a 100D zero-vector.

2.2.2. Protein Network Data

The protein network data we downloaded is scored links between proteins. The higher the score, the greater the probability of interactions between proteins. We filtered all scored links with 400 points, leaving only scored links whose score higher than 400, and then integrated the filtered protein network data into a PPI scored matrix. Each row of this matrix is a vector that represents the interaction of a protein with other proteins. If protein A interacts with another protein B in selected data, we set the value at the corresponding position in the vector to the fraction of these two proteins; otherwise, we set it to 0.

2.2.3. Protein Domain Data

In proteins, the types and number of domains and the relative positions of different domains will affect the functions of the protein. To fully discover and extract the comprehensive information of the type, number, and position of domains in proteins to improve the performance of the model, we first need to sort the domains contained in each protein according to the information of positions in the domain data, so that we can obtain the information relative positions of different domains. However, the position information given by the database is only a possible range of domains in the protein sequence. For example, if the database provides the position of domain D in the sequence of protein P is 60–200, this only indicates that a domain D exists in the area of 60–200 in protein P, but we cannot obtain the actual length and location of this domain D. This is the result of technical limitations, which cause the existence of different domains to overlap, even a region completely contains another region, in a protein, and makes it challenging to sort domains.

In our experiments, we proposed a simple sorting method based on regional center points to solve this problem. Specifically, in a specific protein, there are three possibilities for the geographical relationship between any two different domains: detached, crossing, and containing. If the relationship is detached, we can quickly sort the two domains. If it is a cross-relationship or a containing-relationship, we calculated the center points of the two regions separately, and then put the domain with a forward center point in front of another one. After this, the information on the type, quantity, and relative position of the domain in the protein are obtained. Next, we learned from the idea of Natural Language Processing and treat each domain as a biological word, so the information of domains describing a specific protein is a biological sentence composed of some domain words in a particular order, while the functions of a protein are what the biological sentence means. The purpose of the domain module is to receive the biological sentence of protein and then abstract the features that represent the meaning of the sentence. Because the number of domains contained in different proteins is inconsistent, here we also need to solve the problem of the inconsistent size of model input. We obtained the maximum number of domains of proteins and used this maximum number (357 for Human and 41 for Yeast) as a standard and proteins with fewer domains than the maximum number were padded with 0. We encoded domains by word Embedding to input it into the model. Specifically, we utilized PyTorch's Sparse layer, which can initialize a simple lookup table to map sparse vectors to dense vectors, to generate a fixed lookup table for the domains. In this lookup table, each domain is represented by a 128-dimensional vector. In principle, the Sparse layer automatically maps high-dimensional one-hot vectors to low-dimensional dense vectors and provides the index of the dense vectors. The dimensions of both the one-hot vectors and the dense vectors are manually set by the user as needed, and we could get the required dense vector by entering the index. Therefore, the domains sentence of Human is represented by a 357*128 two-dimensional matrix, while the domains sentence of Yeast is represented by a 41*128 two-dimensional matrix. The Sparse layer will be integrated into the model and trained together, that is, as the model is continuously optimized, the representation vectors of domains in the lookup table will become increasingly accurate.

2.2.4. Protein GO Terms

Given that a large number of specific GO terms often only exist in the annotation sets of a small number of proteins (You et al., 2018), and considering the calculation limit, we ranked the GO terms according to the number of annotations in proteins, and then use a set of thresholds (40 for BP, 20 for MF and 20 for CC) to select the GO terms, which contains 491 BP terms, 321 MF terms, and 240 CC terms, for Human, and a set of thresholds (10 for BP, 10 for MF and 10 for CC) to select the GO terms, which contains 373 BP terms, 171 MF terms, and 151 CC terms, for Yeast. We created three binary vectors for each protein to represent the labels of three sub-ontologies of GO: BP Ontology, MF Ontology, and CC Ontology. If a protein is annotated by a GO term, the value at the corresponding position of the label vector is set as 1, and otherwise is set as zero. Please note that all GO categories in the label vectors are selected.

2.3. Deep Model

We trained three models for the three sub-ontologies of GO. We randomly extracted 80% of the training data for iterative training of the model, and used the remaining 20% to verify the performance of the model after each iteration, and retained the model with the best generalization performance. Given that our model needs to receive input from three aspects of sequence, domain content, and PPI network information, as shown in **Figure 1**, we divided the model into four components: Sequence sub-model, Domain sub-model, PPI-Net sub-model, and Weighted Classifier.

2.3.1. Sequence Sub-model

The input of this sub-model is a two-dimensional 3-grams-vector-matrix that represents protein sequence information. To extract in-depth high-dimensional features of protein biological sequences, we design and implement a model based on convolutional neural networks (CNN). The neural network is a mathematical algorithm model that mimics the behavioral characteristics of biological neural networks for distributed and parallel information processing (Haykin, 1994). In CNN, there is depth structure, and the input is convolved to obtain the output (LeCun et al., 1998), the convolution layer contains multiple convolution kernels, which can make the model extract more features in different aspects. In our experiment, we used a 1-Dimensional convolutional neural network, which uses a one-dimensional convolution kernel to perform convolution operations on the input data. After the sequence input is convolved to extract features, the output feature map is passed to the pooling layer for feature selection and information filtering; this is because the feature map still contains redundancy. Here, we use the max-pooling layer to treat the feature map. After processing, the selected feature map will be passed to the next layer as input. Specifically, three convolutional layers were set for the sequence sub-model, which were connected end to end. The feature map obtained after the convolution operation of each convolutional layer uses a maximum pooling layer to filter information to remove redundancy. The in-channels of the first convolutional layer are the same width as the input sequence information matrix and are set to 100. The in-channels of the other two convolutional layers are the same as the out-channels of the previous layer, and the out-channels of the three convolutional layers are set as 64, 32, and 16, respectively. For

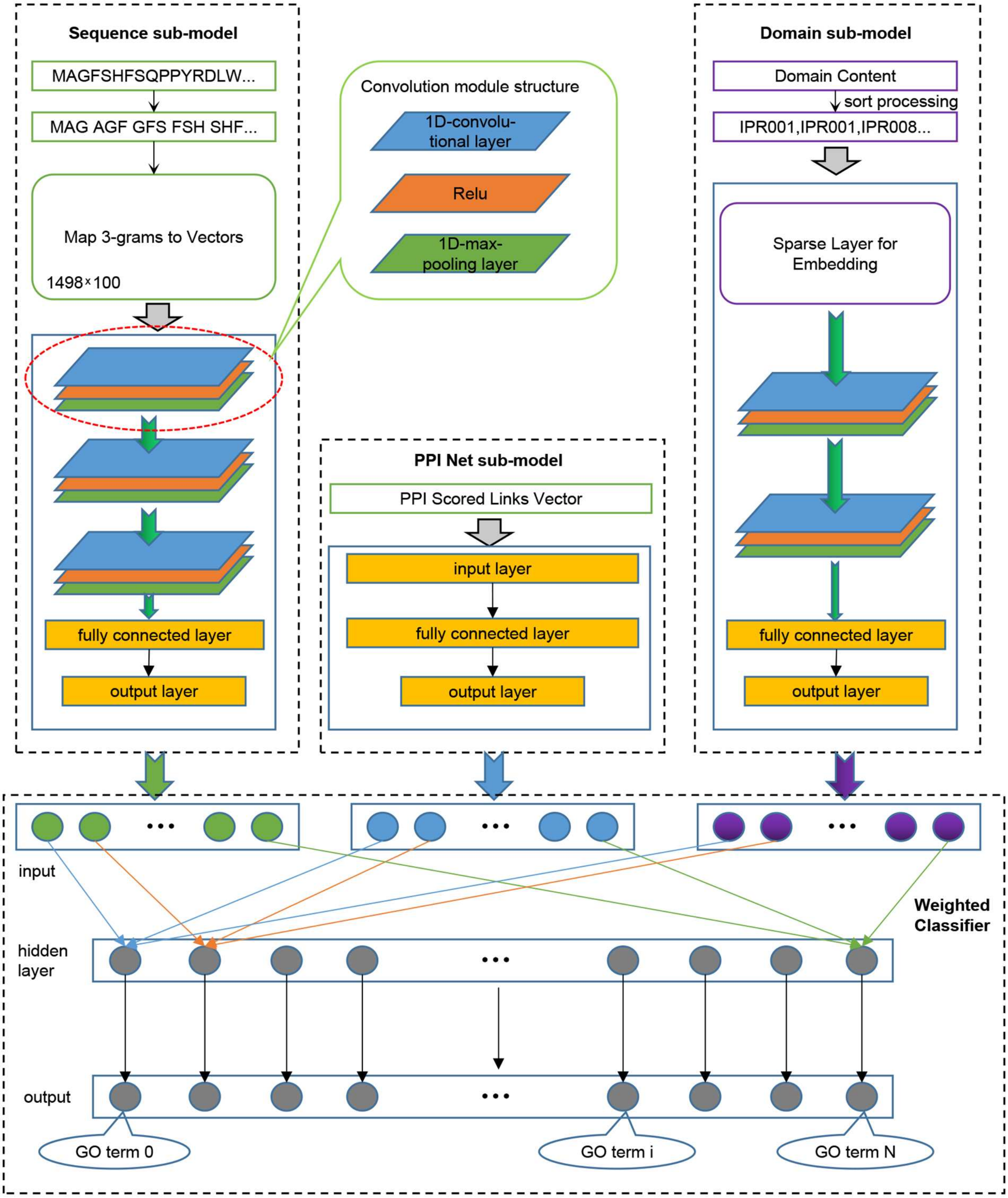


FIGURE 1 | The integrated deep learning model architecture. (1) The Sequence sub-model utilizes 1-Dimensional convolutional neural networks to extract features from sequence input, which was encoded as 3-grams and then mapped to 3-grams-vector-matrix. (2) The PPI Net sub-model is generated to dense the features from PPI Network using classical neural networks. (3) The Domain sub-model initializes a Sparse layer, which is integrated into the sub-model to optimize, to generate a lookup table for domains, and the sorted domains sentence processed by the Sparse layer is entered into 1-Dimensional convolutional neural networks to extract features. (4) All the output features of the three sub-models are combined and entered into the Weighted Classifier, and the output vector represents the probability of GO terms.

each convolution layer, a convolution kernel with a size of 16 is used for the convolution operation with a step size of 1. In order to completely extract the input features, padding was performed on the input with 0 before each convolution. Each maximum pooling layer is filtered using a kernel of size 2 with a step size of 2. The output feature map of the last pooling layer will be tiled into one dimension and input to the fully connected (FC) layers for dimensionality reduction. Finally, a feature vector representing the protein sequence information was obtained. The number of nodes in the output layer of the fully connected layer is set according to the number of three GO sub-ontology. Specifically, for Human, it was set as 491 for BP, 321 for MF, and 240 for CC, and for Yeast, it was set as 373 for BP, 171 for MF, and 151 for CC.

2.3.2. PPI-Net Sub-model

In the PPI scored matrix, the feature vectors that characterize the interaction between proteins and other proteins have large dimensions, which are 18,901 for Human and 6,054 for Yeast, respectively, so we built a three-layer trapezoidal neural network module to dense the PPI features. In this module, the number of nodes in the input layer is the same as the dimension of the input feature vector, which is 18,901 for Human and 6,054 for Yeast. The number of nodes in the hidden layer is set to an intermediate value according to the number of nodes in the input layer and the output layer, which are 4,096 for Human and 2,048 for Yeast. And the size of the output layer is based on different species and GO sub-ontology, and is the same as the output layer of the Sequence sub-model.

2.3.3. Domain Sub-model

The input of the Domain sub-model is the sorted protein domain content information. According to the input data, the first structure of the module is the integrated Sparse layer, the number of embedding is 14,243 for Human, and 6,708 for Yeast, and embedding dim are set as 128. For a specific protein, the output of the Sparse layer of the domain sentence input is a two-dimensional matrix. Therefore, similar to the sequence sub-model, we constructed a convolutional neural networks module containing two 1-D convolutional layers and two max-pooling layers. The in-channels of the first convolutional layer are set to 357 for Human, and 41 for Yeast, the in-channels of the second convolutional layer are consistent with the out-channels of the previous layer, and the out-channels of the two convolutional layers are set to 128 and 64. Besides, each convolutional layer used a convolution kernel of size 2 to perform a convolution operation with a step size of 2. In order to completely extract the input features, we padded the input with 0 before each convolution. The setting of the two maximum pooling layers is the same as the setting of the maximum pooling layer in the Sequence sub-model. The feature map output by the last pooling layer is tiled into one dimension and then input to the fully connected layers to reduce the dimension and the output layer of the fully connected layer. The size of the output layer is based on different species and GO sub-ontology, and is the same as the output layer of the Sequence sub-model.

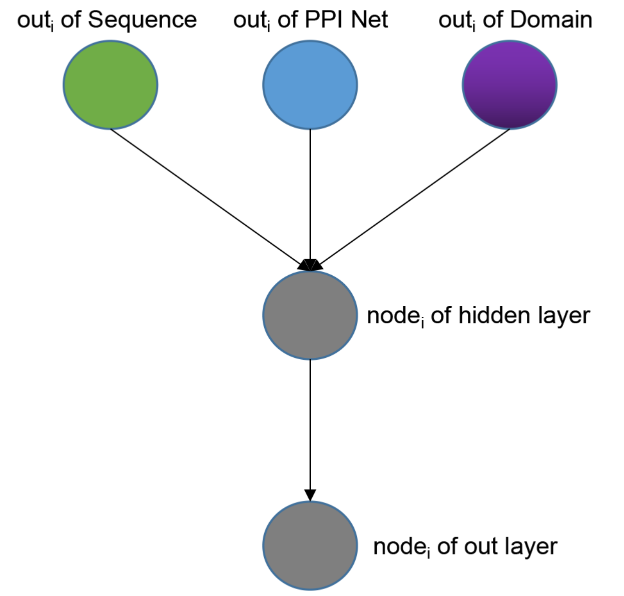


FIGURE 2 | The architecture of one single GO classifier in the weighted classifier.

2.3.4. Weighted Classifier

Weighted Classifier accepts output vectors from three sub-models: Sequence sub-model, Domain sub-model, PPI-Net sub-model. Through training, each GO classifier learns and optimal the weights that receive the features from three sub-models to achieve the best effect of multi-label classification. Note that the output vectors of the three modules have the same dimensions. As a whole, our Weight Classifier is a three-layer non-fully connected network model. The number of nodes in the input layer is the sum of the number of output nodes of the three sub-models, and both the nodes of hidden layer and the nodes of out layer are the same as nodes of the output layer of the three sub-models, which are set according to different species and GO sub-ontology. From the perspective of a single GO classifier, the structure is shown in **Figure 2**. For a specific GO classifier, the hidden node only accepts three features, which are from the corresponding position of the output vector of three sub-model, respectively, corresponding to the GO category, and to extract the corresponding area, we used a binary mask matrix to implement this connection control. The output node of the Classifier also only receives the output of the corresponding hidden node, and we also used a binary mask matrix to implement connection control. In general, let the entire Weight Classifier as a whole again, each node in the hidden layer is only connected to the three corresponding nodes in the output layer, and each node in the output layer is connected to only one corresponding hidden layer node. Therefore, the weights between the hidden layer nodes and the input layer nodes represent the preference of the Classifier for features from three sub-models, and the weights between the output layer nodes and hidden layer nodes globally balance the output values of the Classifier to the same level.

For all components of the model, we used the Rectified-linear-unit (ReLU) (Glorot et al., 2011), which could improve the computational efficiency and retain gradient (Nair and Hinton, 2010), as the activation function. Besides, by running specific optimization algorithms to minimize the loss function, the DNN model can be iteratively optimized by updating the weights and biases. Especially, the model is trained using an adaptive optimizer, Adam (Kingma and Ba, 2014).

2.4. Evaluate Methods

We evaluate the performance of the model through three measures, which are F-max, AUPR (area under the precision-recall curve), and AUC (area under the receiver operator characteristics curve), where F-max and AUC are used in the CAFA challenge (Radivojac et al., 2013). We use the standard provided by CAFA to calculate F-max and the formulas as follows:

$$F_{max} = \max_t \{\frac{2 \cdot pr(t) \cdot rc(t)}{pr(t) + rc(t)}\} \tag{1}$$

where $pr(t)$ and $rc(t)$, respectively represent precision and recall of the threshold $t \in [0, 1]$, and can be calculated by the following formulas:

$$pr(t) = \frac{1}{m(t)} \cdot \sum_{i=1}^{m(t)} pr_i(t) \tag{2}$$

and

$$rc(t) = \frac{1}{n} \cdot \sum_{i=1}^{n} rc_i(t) \tag{3}$$

where $m(t)$ is the number of proteins that annotated with at least one GO term using a threshold t, n is the total number of proteins in the target data set. $pr_i(t)$ and $rc_i(t)$ represent the precision and recall of a specific protein i using a threshold t, and are calculated by the following formulas:

$$pr_i(t) = \frac{\sum_f I(f \in P_i(t) \wedge f \in T_i)}{\sum_f I(f \in P_i(t))} \tag{4}$$

and

$$rc_i(t) = \frac{\sum_f I(f \in P_i(t) \wedge f \in T_i)}{\sum_f I(f \in T_i)} \tag{5}$$

where f is a functional term in the ontology, Function $I(\cdot)$ is the standard indicator function. T_i is the set of true labels for protein i, and $P_i(t)$ is the set of predicted labels for protein i using a threshold t. Once the precision and recall that calculated by different values of t for a particular functional term were determined overall proteins, we could then calculate the AUPR using the trapezoid rule. Compared with AUC, AUPR has a greater penalty for false positives[6].

We also calculate the AUC value for each model of the GO sub-ontology, and the calculation formulas are as follows:

$$AUC = \int_{-\infty}^{\infty} TPR(t)(-FPR(t))dt, \tag{6}$$

$$TPR(t) = \frac{TP(t)}{TP(t) + FN(t)} \tag{7}$$

and

$$FPR(t) = \frac{FP(t)}{FP(t) + TN(t)} \tag{8}$$

where TP is the number of true positives, FP is the number of false positives, and TN is the number of true negatives, FN is the number of false negatives.

2.5. Model Implementation and Computing Environment

We used PyTorch, a Python-based deep learning framework, to implement our model. To speed up the training process, we used a *RHEL* server with four *NVIDIACorporationGM*107*GL* graphics cards installed and total video memory of 32 GB. Under a set of parameters, the whole training time for the most computationally-intensive BP model is <10 h. In terms of prediction, in the case where the sequence, domain, and PPI input information of the predicted protein has been processed in advance, using an optimized model to predict 1,000 proteins takes about 6 min.

3. RESULTS

3.1. Experiment

Owing to the complexity of our model composition and the requirement to determine a large number of hyperparameters, we first pre-trained the three-component sub-models of Sequence, Domain, and PPI Net. We used the GO annotations of proteins as a label and calculated the binary cross-entropy between the predicted values and the actual values, and use this as the loss to back-propagate to update the weights and biases between the nodes connected in the model. We manually adjusted the hyper-parameters, such as the learning rate and batch-size of each module, and selected the optimal model based on the validation loss value using the training set. After adjusting the parameters of the three sub-modules, we used the output of these three fine-tuned models as input to manually adjusted the hyperparameters of the Weighted Classifier, and also select the optimal model based on the validation loss value using the training set. **Tables S1–S4** shows the details of the training of different hyperparameters.

We used 5-fold cross-validation on the training set to test the performance of the model, and the results are shown in **Table 1**. It is clear that the model has achieved a favorable F-max value for each sub-ontology of GO, which indicates that our method is an effective protein function prediction method.

3.2. Evaluating the Performance of Using Domain Content

Using the comprehensive information of types, quantities, and positions of protein domain content for prediction of protein function is the crucial component and emphasis of this research. In order to explore and explain the critical role of comprehensive domain information on protein function prediction, the deep models without the domain module were constructed for three sub-ontology of GO, and each model contained only the Sequence sub-model, PPI-Net sub-model, and Weighted Classifier, and we named it SN2GO. Among SN2GO, since the Sequence sub-model and PPI-Net sub-model in the SDN2GO model are pre-trained separately, the structure and hyperparameter settings of the Sequence sub-model and the PPI-Net sub-model are the same as those of the corresponding modules in the SDN2GO model, and the Weighted Classifier removes the relevant part of the domain from the input layer, the settings of the hidden layer and output layer are still the same as those of the SDN2GO Weighted Classifier. To ensure fairness of comparison, we also manually readjusted the learning rate and batch size hyperparameters and selected the optimal Weighted Classifier model for SN2GO.

TABLE 1 | The 5-fold cross validation results of training data.

Method	BP			MF			CC		
	F_{max}	AUPR	AUC	F_{max}	AUPR	AUC	F_{max}	AUPR	AUC
SN2GO (human)	0.473	0.441	0.908	0.546	0.527	0.938	0.587	0.600	0.949
SDN2GO (human)	**0.507**	**0.487**	**0.921**	**0.653**	**0.655**	**0.957**	**0.601**	**0.617**	**0.952**
SN2GO (yeast)	0.414	0.289	0.810	0.548	0.435	0.870	0.520	0.395	**0.881**
SDN2GO (yeast)	**0.415**	**0.304**	**0.839**	**0.611**	**0.530**	**0.903**	**0.528**	**0.424**	0.878

The bold values indicate the best values.

We observed the performance of SN2GO on the training set and compared it with SDN2GO. As the same, we used SN2GO to perform a 5-fold cross-validation experiment on the training set. **Table 1** shows the cross-validation results of SN2GO. We find that compared with SN2GO, the performance of the SDN2GO that uses domain information has been significantly improved on all the sub-ontology of GO, especially in the MF Ontology of humans, the F-measure value of SDN2GO has been enhanced by nearly 20% (0.65 vs. 0.55) compared to SN2GO. As shown in **Figure 3**, the PR curves of SDN2GO and SN2GO on validation data of humans, it is clear that the red PR curve surrounds the other one on each sub-ontology. This result shows that domain

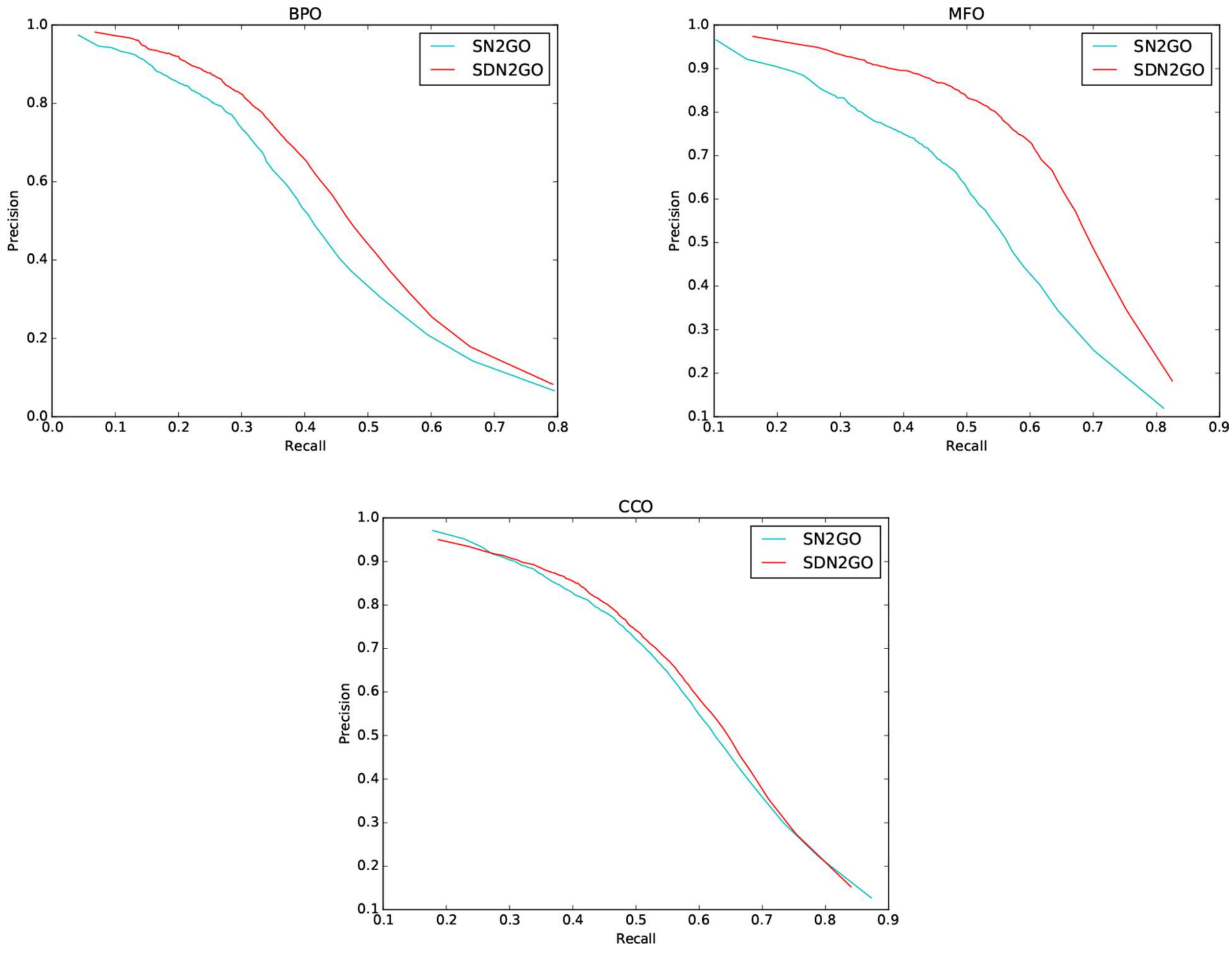


FIGURE 3 | Precision-recall (P-R) curves of SDN2GO and SN2GO. The performances of the two methods were evaluated on the validation data of human in each sub-ontology of GO (gene ontology).

information plays an essential role in protein function prediction, and proves that our coding and processing methods for protein domain information and the sub deep learning models for domains are useful and meaningful.

3.3. Comparison With Competing Methods

In order to further verify the performance of SDN2GO, we compared the two novel methods, NetGO and DeepGO, on the independent test set. Both of these two methods are competitive and excellent in protein function prediction and have achieved outstanding results on some datasets. As a state-of-the-art machine learning method for protein function prediction, NetGO provides constructive ideas on how to integrate features based on different sources. At the same time, DeepGO is quite representative of using deep learning technology for protein function prediction. Specifically, NetGO integrates five different types of sequence-based evidence and massive network information into the learning to rank (LTR) framework to predict protein function. We uploaded the protein sequence of the independent test set in Fasta format to the AFP (automated function prediction) webserver (http://issubmission.sjtu.edu.cn/netgo/) released by NetGO and then downloaded the prediction result of NetGO in txt format after a while. DeepGO uses convolutional neural networks to extract protein sequence features and combines known PPI network information as combined features to predict protein functions. We downloaded all source code of DeepGO from GitHub and downloaded the required data, and the fine turned neural network models saved in PKL format from the provided webserver (http://deepgo.bio2vec.net/data/deepgo/), and then entered the test protein sequence in Fasta format to this open-source tool, and obtained the prediction results of DeepGO. Besides, the BLAST was also used in comparative experiments.

The comparison results are shown in **Table 2**. We have observed that BLAST performs well on every GO sub-ontology, which illustrates again that the sequence homology-based BLAST method is still quite competitive. NetGO and DeepGO performed well on MFO and BPO, respectively, but did not achieve their claimed effects on other sub-ontology. We further analyzed the prediction results of these two methods, and we found that the false-positive rates of both of them are relatively high, which leads to their inability to obtain high precision values. **Figure 4**, which shows the PR curves of MFO on independent test sets for various methods, demonstrates our analysis results from one aspect. The PR curves of BPO and CCO and other specific details can be seen in **Figures S1, S2**. Obviously, SDN2GO outperformed other methods on all sub-ontologies, especially on MFO. Those shows that our model has excellent generalization performance and is a currently competitive method for protein function prediction. In particular, we paid attention to the performance of SN2GO, which lacks the domain sub-model on the test set. The results show that its performance on BPO and MFO is far worse than that of SDN2GO, and prove that extracting features from protein domains for protein function prediction is feasible, and will improve the accuracy of GO term labeling for proteins, especially on BPO and MFO.

TABLE 2 | The comparison results of the competing method on the independent testing set.

Method	BP			MF			CC		
	F_{max}	AUPR	AUC	F_{max}	AUPR	AUC	F_{max}	AUPR	AUC
BLAST	0.347	0.192	0.771	0.381	0.292	0.873	0.386	0.245	0.860
DeepGO	0.321	0.095	0.729	0.291	0.117	0.784	0.210	0.080	0.687
NetGO	0.173	0.048	0.594	0.386	0.243	0.919	0.217	0.092	0.669
SN2GO	0.132	0.044	0.893	0.423	0.306	0.953	0.384	0.264	**0.948**
SDN2GO	**0.361**	**0.203**	**0.917**	**0.561**	**0.471**	**0.964**	**0.432**	**0.290**	0.947

The bold values indicate the best values.

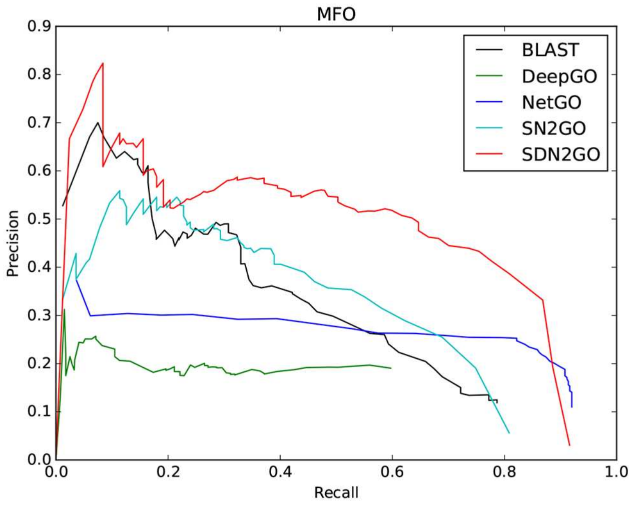


FIGURE 4 | Precision-recall (P-R) curves of BLAST, DeepGO, NetGO, SN2GO, and SDN2GO. The performances of the five methods were evaluated on the independent testing set in MFO (molecular function ontology).

4. DISCUSSION

SDN2GO, an integrated deep learning-based weight model we have proposed, combines three aspects of information: protein sequence, protein domain content, and known protein-protein interaction networks. We constructed three sub-models for these three aspects of information, and then learned and extracted three components of features through pre-training the sub-models. Each GO term of the protein was finally scored and annotated through the integrated deep learning weight classifier. The 5-fold cross-validation results show that SDN2GO is a stable and reliable method for protein function prediction. In order to further verify the generalization performance and competitiveness of SDN2GO, we constructed an independent test set based on the principle of time-delay for comparison with the novel method and the classic BLAST method. The comparison results show that our method has achieved the maximum F-max value for each sub-ontology of GO.

Many studies illustrated that protein sequence and PPI network are valid for protein function (Kirac and

Ozsoyoglu, 2008; Jiang and McQuay, 2011; Nguyen et al., 2011; Baryshnikova, 2016; Kulmanov et al., 2018). Besides, some researchers have used protein domain information to predict protein function (Altshul, 1997; Forslund and Sonnhammer, 2008), but they only focused on a single aspect of type or structure of the domain and failed to fully mine the general characteristics of various aspects of the domain. We considered this and drowned lessons from the principle of NLP to encode domains to integrate the type, quantity, and position information of the protein domains, and utilized the convolutional neural network to extract the general characteristics of the domains, which is the advantage of our model. We built a comparison model SN2GO based on SDN2GO without domain sub-model and conducted comparative experiments on both the training data and the independent test set. The results show that the domain information has significantly improved the prediction effect of the model, especially in BPO On MFO; this might be because the domain information, as a higher-level protein feature than sequence, is more intuitive in expression and closer to the functions of the protein. And to a certain extent, the comparison results illustrated the correctness and generalizability of our methods of protein domain information processing and feature extraction.

In the future, we will continue to improve our model, such as adding more GO annotation categories to expand the scale of multi-label classification. Besides, we will also try to integrate more aspects of protein-related features, such as protein structure information and co-expression information, into our model to explore the role of different information on protein function prediction.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/Charrick/SDN2GO/tree/master/data.

AUTHOR CONTRIBUTIONS

YC and LD conceived this work and designed the experiments. YC and JW built the experimental environment. YC carried out the experiments. YC, LD, and JW collected the data and analyzed the results. YC and LD wrote, revised, and approved the manuscript.

FUNDING

This research was funded by National Natural Science Foundation of China under grant Nos. 61972422 and 61672541.

ACKNOWLEDGMENTS

We would like to thank the Experimental Center of School of Computer Science of Central South University, for providing computing resources.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fbioe.2020.00391/full#supplementary-material

REFERENCES

Altshul, S. F. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*. 25, 3389–3402. doi: 10.1093/nar/25.17.3389

Asgari, E., and Mofrad, M. R. (2015). Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLoS ONE* 10:e0141287. doi: 10.1371/journal.pone.0141287

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., et al. (2000). Gene ontology: tool for the unification of biology. *Nat. Genet*. 25, 25–29. doi: 10.1038/75556

Barrell, D., Dimmer, E., Huntley, R. P., Binns, D., O'Donovan, C., and Apweiler, R. (2009). The goa database in 2009-an integrated gene ontology annotation resource. *Nucleic Acids Res*. 37, D396–D403. doi: 10.1093/nar/gkn803

Baryshnikova, A. (2016). Systematic functional annotation and visualization of biological networks. *Cell Syst*. 2, 412–421. doi: 10.1016/j.cels.2016.04.014

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., et al. (2009). Blast+: architecture and applications. *BMC Bioinformatics* 10:421. doi: 10.1186/1471-2105-10-421

Chen, Y., Li, Y., Narayan, R., Subramanian, A., and Xie, X. (2016). Gene expression inference with deep learning. *Bioinformatics* 32, 1832–1839. doi: 10.1093/bioinformatics/btw074

Consortium, U. (2015). Uniprot: a hub for protein information. *Nucleic Acids Res*. 43, D204–D212. doi: 10.1093/nar/gku989

Consortium, U. (2019). Uniprot: a worldwide hub of protein knowledge. *Nucleic Acids Res*. 47, D506–D515. doi: 10.1093/nar/gky1049

Dao, F.-Y., Lv, H., Zulfiqar, H., Yang, H., Su, W., Gao, H., et al. (2020). A computational platform to identify origins of replication sites in eukaryotes. *Brief. Bioinform. [Preprint]* bbaa017. doi: 10.1093/bib/bbaa017

Federhen, S. (2012). The ncbi taxonomy database. *Nucleic Acids Res*. 40, D136–D143. doi: 10.1093/nar/gkr1178

Forslund, K., and Sonnhammer, E. L. (2008). Predicting protein function from domain content. *Bioinformatics* 24, 1681–1687. doi: 10.1093/bioinformatics/btn312

Gillis, J., and Pavlidis, P. (2013). Characterizing the state of the art in the computational assignment of gene function: lessons from the first critical assessment of functional annotation (cafa). *BMC Bioinformatics* 14:S15. doi: 10.1186/1471-2105-14-S3-S15

Glorot, X., Bordes, A., and Bengio, Y. (2011). "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (Fort Lauderdale, FL), 315–323.

Hakala, K., Kaewphan, S., Björne, J., Mehryary, F., Moen, H., Tolvanen, M., et al. (2019). Neural network and random forest models in protein function prediction. *BioRxiv* 690271. doi: 10.1101/690271

Hamp, T., Kassner, R., Seemayer, S., Vicedo, E., Schaefer, C., Achten, D., et al. (2013). Homology-based inference sets the bar high for protein function prediction. *BMC Bioinformatics* 14:S7. doi: 10.1186/1471-2105-14-S3-S7

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ: Prentice Hall PTR.

Heddad, A., Brameier, M., and MacCallum, R. M. (2004). "Evolving regular expression-based sequence classifiers for protein nuclear localisation," in *Workshops on Applications of Evolutionary Computation* (Berlin; Heidelberg: Springer), 31–40. doi: 10.1007/978-3-540-24653-4_4

Hunter, S., Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Binns, D., et al. (2009). Interpro: the integrative protein signature database. *Nucleic Acids Res.* 37, D211–D215. doi: 10.1093/nar/gkn785

Jiang, J. Q., and McQuay, L. J. (2011). Predicting protein function by multi-label correlated semi-supervised learning. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 9, 1059–1069. doi: 10.1109/TCBB.2011.156

Jones, P., Binns, D., Chang, H.-Y., Fraser, M., Li, W., McAnulla, C., et al. (2014). Interproscan 5: genome-scale protein function classification. *Bioinformatics* 30, 1236–1240. doi: 10.1093/bioinformatics/btu031

Kahanda, I., and Ben-Hur, A. (2017). "Gostruct 2.0: Automated protein function prediction for annotated proteins," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics* (New York, NY), 60–66. doi: 10.1145/3107411.3107417

Kawashima, S., and Kanehisa, M. (2000). Aaindex: amino acid index database. *Nucleic Acids Res.* 28, 374–374. doi: 10.1093/nar/28.1.374

Kiemer, L., Bendtsen, J. D., and Blom, N. (2005). Netacet: prediction of n-terminal acetylation sites. *Bioinformatics* 21, 1269–1270. doi: 10.1093/bioinformatics/bti130

Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv [Preprint] arxiv*:1412.6980.

Kirac, M., and Ozsoyoglu, G. (2008). "Protein function prediction based on patterns in biological networks," in *Annual International Conference on Research in Computational Molecular Biology* (Berlin: Heidelberg: Springer), 197–213. doi: 10.1007/978-3-540-78839-3_18

Kulmanov, M., and Hoehndorf, R. (2020). Deepgoplus: improved protein function prediction from sequence. *Bioinformatics* 36, 422–429. doi: 10.1101/615260

Kulmanov, M., Khan, M. A., and Hoehndorf, R. (2018). Deepgo: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics* 34, 660–668. doi: 10.1093/bioinformatics/btx624

Lai, H.-Y., Zhang, Z.-Y., Su, Z.-D., Su, W., Ding, H., Chen, W., et al. (2019). iproep: a computational predictor for predicting promoter. *Mol. Ther. Nucleic Acids* 17, 337–346. doi: 10.1016/j.omtn.2019.05.028

Lanchantin, J., Singh, R., Lin, Z., and Qi, Y. (2016). Deep motif: Visualizing genomic sequence classifications. *arXiv [Preprint] arxiv*:1605.01133.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Lv, H., Zhang, Z.-M., Li, S.-H., Tan, J.-X., Chen, W., and Lin, H. (2019a). Evaluation of different computational methods on 5-methylcytosine sites identification. *Brief. Bioinform.* doi: 10.1093/bib/bbz048

Lv, Z., Ao, C., and Zou, Q. (2019b). Protein function prediction: from traditional classifier to deep learning. *Proteomics* 19:1900119. doi: 10.1002/pmic.201900119

Nair, V., and Hinton, G. E. (2010). "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (Haifa), 807–814.

Nguyen, C. D., Gardiner, K. J., and Cios, K. J. (2011). Protein annotation from protein interaction networks and gene ontology. *J. Biomed. Inform.* 44, 824–829. doi: 10.1016/j.jbi.2011.04.010

Oliver, S. (2000). Guilt-by-association goes global. *Nature* 403, 601–602. doi: 10.1038/35001165

Pierleoni, A., Martelli, P. L., and Casadio, R. (2008). PredGPI: a GPI-anchor predictor. *BMC Bioinformatics* 9:392. doi: 10.1186/1471-2105-9-392

Piovesan, D., and Tosatto, S. C. (2019). INGA 2.0: improving protein function prediction for the dark proteome. *Nucleic Acids Res.* 47, W373–W378. doi: 10.1093/nar/gkz375

Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T., Sokolov, A., et al. (2013). A large-scale evaluation of computational protein function prediction. *Nat. Methods* 10, 221–227. doi: 10.1038/nmeth.2340

Richardson, J. S. (1981). The anatomy and taxonomy of protein structure. *Adv. Prot. Chem.* 34, 167–339. doi: 10.1016/S0065-3233(08)60520-3

Rifaioglu, A. S., Doğan, T., Martin, M. J., Cetin-Atalay, R., and Atalay, V. (2019). Deepred: automated protein function prediction with multi-task feed-forward deep neural networks. *Sci. Rep.* 9, 1–16. doi: 10.1038/s41598-019-4 3708-3

Schwikowski, B., Uetz, P., and Fields, S. (2000). A network of protein-protein interactions in yeast. *Nat. Biotechnol.* 18, 1257–1261. doi: 10.1038/82360

Sønderby, S. K., Sønderby, C. K., Nielsen, H., and Winther, O. (2015). "Convolutional LSTM networks for subcellular localization of proteins," in *International Conference on Algorithms for Computational Biology* (Springer), 68–80. doi: 10.1007/978-3-319-21233-3_6

Sønderby, S. K., and Winther, O. (2014). Protein secondary structure prediction with long short term memory networks. *arXiv [Preprint] arxiv*:1412.7828.

Spencer, M., Eickholt, J., and Cheng, J. (2014). A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 12, 103–112. doi: 10.1109/TCBB.2014.2343960

Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., et al. (2015). String v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.* 43, D447–D452. doi: 10.1093/nar/gku1003

Tan, J.-X., Li, S.-H., Zhang, Z.-M., Chen, C.-X., Chen, W., Tang, H., et al. (2019). Identification of hormone binding proteins based on machine learning methods. *Math. Biosci. Eng.* 16, 2466–2480. doi: 10.3934/mbe.2019123

Wang, J., Zhang, J., Cai, Y., and Deng, L. (2019a). Deepmir2go: Inferring functions of human micrornas using a deep multi-label classification model. *Int. J. Mol. Sci.* 20:6046. doi: 10.3390/ijms20236046

Wang, L., Liu, Y., Zhong, X., Liu, H., Lu, C., Li, C., et al. (2019b). Dmfold: A novel method to predict rna secondary structure with pseudoknots based on deep learning and improved base pair maximization principle. *Front. Genet.* 10:143. doi: 10.3389/fgene.2019.00143

Weaver, R. (2011). *Molecular Biology (WCB Cell & Molecular Biology)*. New York, NY: McGraw-hill Education.

Wei, L., Ding, Y., Su, R., Tang, J., and Zou, Q. (2018). Prediction of human protein subcellular localization using deep learning. *J. Parallel Distrib. Comput.* 117, 212–217. doi: 10.1016/j.jpdc.2017.08.009

Yang, J., Yan, R., Roy, A., Xu, D., Poisson, J., and Zhang, Y. (2015). The i-tasser suite: protein structure and function prediction. *Nat. Methods* 12:7. doi: 10.1038/nmeth.3213

You, R., Yao, S., Xiong, Y., Huang, X., Sun, F., Mamitsuka, H., et al. (2019). Netgo: improving large-scale protein function prediction with massive network information. *Nucleic Acids Res.* 47, W379–W387. doi: 10.1093/nar/gkz388

You, R., Zhang, Z., Xiong, Y., Sun, F., Mamitsuka, H., and Zhu, S. (2018). Golabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics* 34, 2465–2473. doi: 10.1093/bioinformatics/bty130

Zhang, C., Zheng, W., Freddolino, P. L., and Zhang, Y. (2018). Metago: Predicting gene ontology of non-homologous proteins through low-resolution protein structure prediction and protein-protein network mapping. *J. Mol. Biol.* 430, 2256–2265. doi: 10.1016/j.jmb.2018.03.004

Zhang, M.-L., and Zhou, Z.-H. (2013). A review on multi-label learning algorithms. *IEEE Trans. Knowl. Data Eng.* 26, 1819–1837. doi: 10.1109/TKDE.2013.39

Zhang, S., Zhou, J., Hu, H., Gong, H., Chen, L., Cheng, C., et al. (2016). A deep learning framework for modeling structural features of RNA-binding protein targets. *Nucleic Acids Res.* 44:e32. doi: 10.1093/nar/gkv1025

Zhu, X.-J., Feng, C.-Q., Lai, H.-Y., Chen, W., and Hao, L. (2019). Predicting protein structural classes for low-similarity sequences by evaluating different features. *Knowl. Based Syst.* 163, 787–793. doi: 10.1016/j.knosys.2018.10.007

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

METHODS
published: 25 June 2020
doi: 10.3389/fbioe.2020.00635

Prediction of G Protein-Coupled Receptors With CTDC Extraction and MRMD2.0 Dimension-Reduction Methods

Xingyue Gu[1], *Zhihua Chen*[1]* *and Donghua Wang*[2]*

[1] *Institute of Computing Science and Technology, Guangzhou University, Guangzhou, China,* [2] *Department of General Surgery, Heilongjiang Province Land Reclamation Headquarters General Hospital, Harbin, China*

The G Protein-Coupled Receptor (GPCR) family consists of more than 800 different members. In this article, we attempt to use the physicochemical properties of Composition, Transition, Distribution (CTD) to represent GPCRs. The dimensionality reduction method of MRMD2.0 filters the physicochemical properties of GPCR redundancy. Matplotlib plots the coordinates to distinguish GPCRs from other protein sequences. The chart data show a clear distinction effect, and there is a well-defined boundary between the two. The experimental results show that our method can predict GPCRs.

Keywords: feature extraction, CTD, MRMD2.0, Matplotlib, predict GPCRs

Edited by:
Zhibin Lv,
University of Electronic Science and Technology of China, China

Reviewed by:
Lijun Dou,
Shenzhen Polytechnic, China
Changli Feng,
Taishan University, China

****Correspondence:***
Zhihua Chen
czhgd@gzhu.edu.cn
Donghua Wang
wangdonghua7885@163.com

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *21 April 2020*
Accepted: *26 May 2020*
Published: *25 June 2020*

Citation:
Gu X, Chen Z and Wang D (2020) Prediction of G Protein-Coupled Receptors With CTDC Extraction and MRMD2.0 Dimension-Reduction Methods.
Front. Bioeng. Biotechnol. 8:635.
doi: 10.3389/fbioe.2020.00635

INTRODUCTION

G protein-coupled receptors (GPCRs) are the largest receptor superfamily. According to their sequence similarity, they are divided into 6 subfamilies (AF), of which the Rhodopsin or rhodopsin-like family is the largest and most widely studied family (Fredriksson et al., 2003; Liu and Zhu, 2019; Ru et al., 2020). Class A has approximately 284 members in humans, and Class B subfamilies can be further divided into two unused families: Class B1, named secretin, secrete protein-like receptors, and Class B2 (adhesion) adhere to GPCRs. Class B1 and Class B2 contain 15 members and 33 members in humans, respectively. The adhesive G protein-coupled receptor (ADGR) family is one of the oldest GPCR families. It exists in primitive animals, and even in several basic fungi, and is the ancestor of the B1 subfamily of GPCRs (Nordstrm et al., 2009; Krishnan et al., 2012). Finally, the class C glutamate family is composed of peptide receptors. The class F frizzled protein family has appsroximately 11 members in humans.

Protein classification is one of the key issues in bioinformatics and plays an important role in the identification and study of gene markers (Tibshirani, 1996; Cheng and Hu, 2018; Feng, 2019; Guo et al., 2019). With the development of machine learning, protein classification and prediction have entered a new era. Machine learning can use previous experience and data to automatically improve the performance of algorithms, build appropriate models, and discriminate new protein sequences. Islam et al. (2017) applied a natural language processing N-Gram model to classify proteins. The above machine learning methods have achieved certain effects in protein classification. This article uses feature extraction and dimension reduction of GPCR proteins to distinguish between the properties of the extracted proteins. Finally, Matplotlab is used to distinguish GPCRs from non-GPCRs. In the article Prediction of G Protein-Coupled Receptors (Liao et al., 2016), the 188D method is used to extract the protein features, and then cross validation and random forest are used to accurately divide the GPCR and non-gpcr protein sequences. In this paper, the CTD mode (Zou et al., 2013) is used, where C represents the content of each hydrophobic amino acid, T represents the frequency of the divalent peptide, and D represents the amino acid distribution at the five positions of the sequence. After using CTDC feature extraction method, the innovative feature of this experiment is that the redundant features are well-extracted using dimensionality reduction. Finally, the machine learning method and Matplotlib are used to draw a graph that distinguishes GPCRs from non-GPCRs.

MATERIALS AND METHODS

Datasets

1. The original 5027 G protein-coupled receptors (GPCRs) were obtained in fasta format from the database (http://www.UniProt.org/); 2. The initial sequence was pre-processed using the protein clustering programme CDHIT (http://cd-hit.org/) to improve the analysis performance and reduce the homology of the predicted sequence (Zou et al., 2020). The critical value of sequence identity was located at 0.8. Finally, 2,495 GPCR sequences were obtained from the positive data set. 3. The positive sequences of all the protein sequences were removed, and 10,386 non-GPCR protein sequences were produced as the positive dataset (Liao et al., 2016).

Feature Extraction Methods

Principle

CTD represents the composition, transition, and distribution, respectively. Its principle is to replace the amino acid sequence with mathematical symbols representing physical and chemical properties (Cheng et al., 2018a). Because the protein sequence information is of different lengths, CTD is used to obtain fixed-length information from proteins as input to machine learning. In protein or peptide sequences, CTD represents physicochemical properties or amino acid distribution patterns of specific structures (Dubchak et al., 1995, 1999; Cai et al., 2003; Zhang et al., 2011; Ding et al., 2017). These features are very important for protein sequence analysis (Wei et al., 2018; Liu et al., 2019; Liu et al., 2019a; Yan et al., 2019; Chen et al., 2020). According to the main amino acid indicators of Tomii and Kanehisa (Kentaro and Minoru, 1996), amino acids are divided into three groups according to seven physical and chemical properties, as shown in **Table 1**.

CTD (Dubchak et al., 1999) is very helpful for enzyme prediction. Composition (Cai et al., 2003; Han et al., 2004; Chen W. et al., 2019; Liu, 2019) refers to the number of specific amino acids in a protein sequence divided by the total length N of the amino acid in the protein sequence:

$$\text{Composition(e)} = \frac{n_e}{N} \tag{i}$$

where n_e represents the sum of the number of e, a particular amino acid, in the sequence. e could be 1, 2, or 3, which represents the type of amino acid.

TABLE 1 | Seven types of physicochemical properties and the division of amino acids.

Seven types of physicochemical properties	Division: 1	Division: 2	Division: 3
Secondary structure; Amino acids	Helix; M, E, A, K, R, H, L, Q	Strand; W, F, T, V, I, Y, C	Coil; S, D, G, P, N
Hydrophobicity; Amino acids	Polar; N, Q, D, E, K, R	Neutral; Y, P, H, S, T, A, G	Hydrophobicity; M, F, I, L, C, W, V
Normalized van der Waals volume; Amino acids	0–2.78; T, S, P, A, G, D	2.95–94.0; Q, L, V, N, E, I	4.03–8.08; M, H, K, F, R, Y, W
Solvent accessibility; Amino acids	Buried; W, V, I, C, G, F, A, L	Exposed; Q, E, D, N, K, P	Intermediate; H, Y, M, S, P, T
Polarizability; Amino acids	0–1.08; G, A, S, D, T	0.128–120.186; G, P, N, V, E, Q, I, L	0.219–0.409; K, M, H, F, R, Y, W
Charge; Amino acids	Positive; K, R	Neutral; Q, G, H, I, A, N, C, L, M, FP, S, T, W, Y, V	Negative; E, D
Polarity; Amino acids	4.9–6.2; L, I, F, W, C, M, V, Y	8.0–9.2; P, A, T, G, S	10.4–13.0; H, Q, R, K, N, E, D

Assuming two specific amino acids are a and b, transition (T) means the number of ab and ba divided by the length of the protein sequence N-1:

$$\text{Transition(ab + ba)} = \frac{n_{ab} + n_{ba}}{N - 1} \tag{ii}$$

The distribution is the position of a specific amino acid in the protein/the total length of the protein sequence, which represents the chain length at which the first, 25, 50, 100% amino acids of this particular amino acid are located.

For example, take the following protein sequence: DEKRADGSTAGPSTDGNPS. According to **Table 1**, DE is the amino acid sequence of classification 2 under Charge, KR is the amino acid sequence of category 3 under Charge, and ADGST is the amino acid sequence of classification 1 under Polarizability. AGPST is an amino acid sequence of Polarity 2, and DGNPS is the amino acid sequence of classification 1 under the Secondary Structure. Thus, our protein sequence is converted by CTD to 2233111112222211111. The following shows how the protein sequence Composition, Transition, Distribution is calculated (see **Figure 1**).

Composition of category 2: 7/(7 + 2 + 10 = 19)= 36.8%; Composition of category 3: 2/19 = 10.5%; Composition of category 1: 10/19 = 52.6%. Transition (23, 32) = 1/18 = 5.5%; Transition (12, 21) = 2/18 = 11.1%; Transition (13, 31) = 1/18

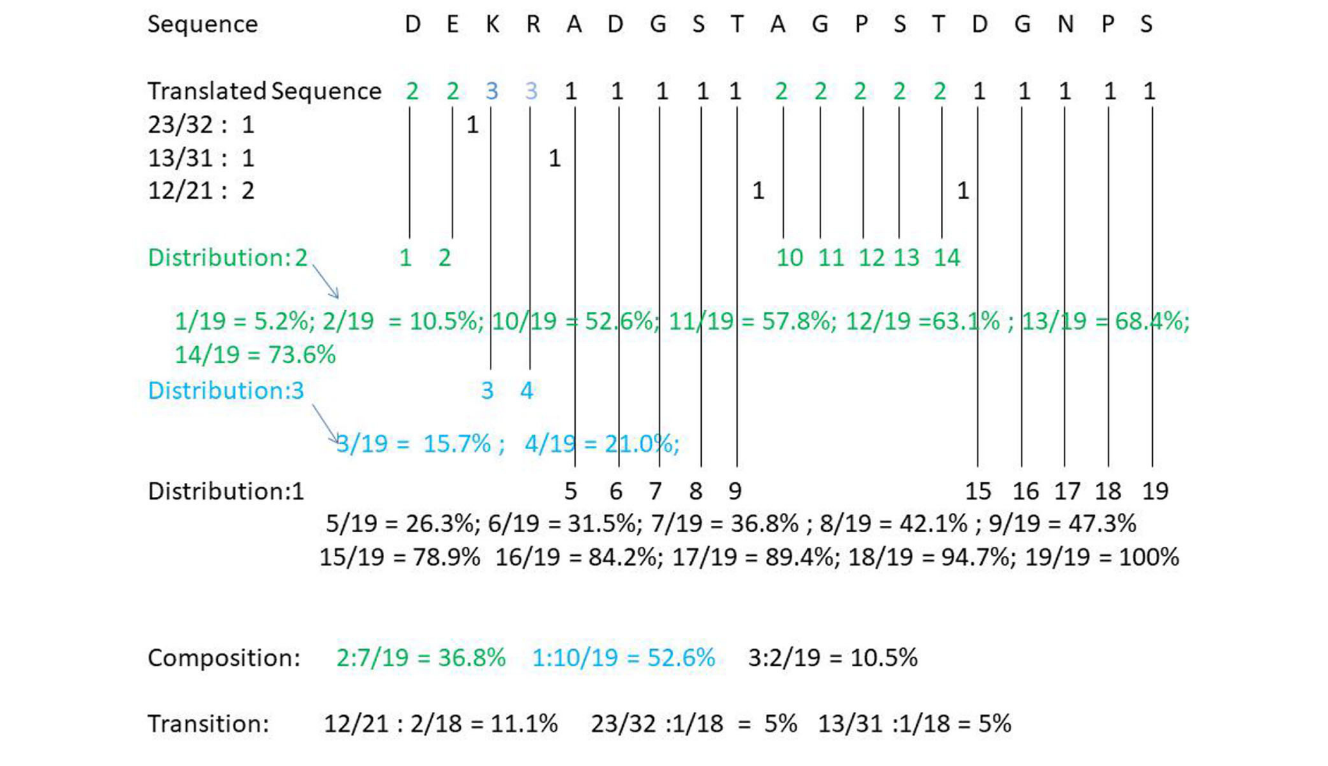


FIGURE 1 | Computational flow of CTD eigenvectors in protein sequences.

= 5.5%. Distribution (1) = 5/19, 6/19, 7/19, 8/19, 15/19, 16/19, 17/19, 18/19, 19/19; Distribution (2) =1/19, 2/19, 10/19, 11/19, 12/19, 13/19, 14/19; Distribution 3 is equal to 3/19, 4/19. The final CTD results of DEKRADGSTAGPSTDGNPS are as follows: Composition (2): 36.8%, Composition (3): 10.5%, Composition (1): 52.6%. T (23, 32): 5.5%, T (12, 21): 11.1%, T (13, 31): 5.5%; D (1): 26.3, 31.5, 36.8, 42.1, 78.9, 84.2, 89.4, 94.7, 100%; D (2): 5.2, 10.5, 52.6, 57.8, 63.1, 68.4, 73.6%; D (3): 15.7, 21.0%.

Dimensionality Reduction

The MRMD2.0 (Wei et al., 2015; Zou et al., 2016a,b) algorithm is used to reduce the dimensions of the files after using CTDC to extract features. The specific process of dimensionality reduction is:

1. Attribute selection: Using analysis of variance to test the significance of the difference between the mean values of two or more samples; maximum correlation and maximum distance MRMD feature classification and accuracy and stability of prediction tasks; MIC is based on a non-parametric information-based maximum parameter exploration for measuring the linear or non-linear strength of two variables X and Y; the minimum absolute contraction and selection operator (LASSO) (Tibshirani, 1996; Guo et al., 2019) uses an L1 regularized linear regression method; Minimal Redundancy-Maximum Correlation (mRMR) method expands the representativeness of a feature set by requiring features to be maximally different from each other; chi-square test is a widely used hypothesis test based on the chi-square distribution for common hypothesis testing; Recursive Feature Elimination (RFE) classifies data according to the size of the correlation coefficients or importance of feature attributes. Through recursive elimination of functions in each cycle, RFE attempts to eliminate possible dependencies and collinearity in the model.
2. Function ranking PageRank algorithm: In the attribute selection method used above, point a to b because feature b is more important than feature a. Finally, the result of each function selection method forms a link list. Using the PageRank algorithm to rank these links, a directed graph is formed, and each feature receives a score. A ranking is then obtained according to the level of the feature, a, b, c, d, e ...
3. Finally, choose the best outcome of the sequence. Since the first feature "a" in the new sequence has the highest score, random forest (Pang et al., 2006; Ding et al., 2016; Cheng et al., 2018b; Liu et al., 2019b; Su et al., 2019; Wei et al., 2019; Xu et al., 2019c; Lv et al., 2020) is used for 5-fold cross-validation starting from the first feature. The highest standard score is made by comparing the three sequences: "a," "a,b;" "a,b,c,d,e." Finally, five data indicators were used: f-score, precision, recall, MCC and AUC (Xu et al., 2018a; Cheng, 2019; Cheng L. et al., 2019; Ding et al., 2019; Zeng et al., 2019a, 2020; Zhang et al., 2019; Liu and Chen, 2020; Wang et al., 2020), and the sequence with the highest index and the highest score for dimension reduction was found. The specific dimension reduction process is shown in **Figure 2**.

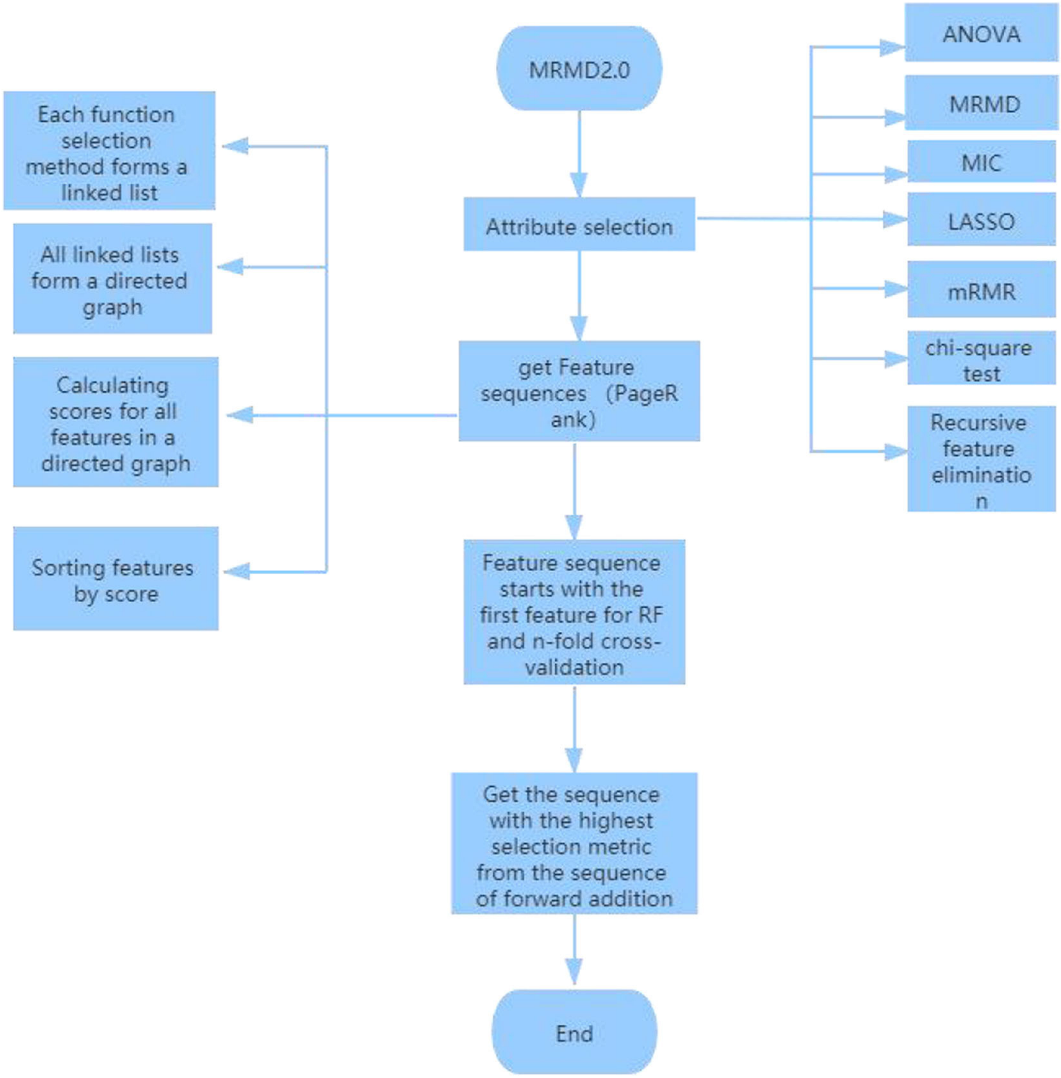

MRMD2.0
Attribute selection
get Feature sequences (PageRank)
Feature sequence starts with the first feature for RF and n-fold cross-validation
Get the sequence with the highest selection metric from the sequence of forward addition
End
Each function selection method forms a linked list
All linked lists form a directed graph
Calculating scores for all features in a directed graph
Sorting features by score
ANOVA
MRMD
MIC
LASSO
mRMR
chi-square test
Recursive feature eliminatio n

FIGURE 2 | The specific dimension reduction process.

Algorithm Steps

GPCR sequence protein features are extracted using specific protein extraction methods. Any two attributes in the extracted features are divided into GPCRs and non-GPCRs. Finally, Matplotlib is used to divide any two attributes in the extracted features into GPCRs and non-GPCRs (the experimental flow chart is shown in **Figure 3**):

(1) Using all the different positive protein samples, extract the corresponding Pfam protein sequence from the “family and domain” of the UniProt website and delete the redundant and identical Pfam number. Then, the unique Pfam number obtained for the positive data set (Liao et al., 2016).

(2) All the protein sequences are integrated into the Pfam number file, and the protein sequences with the same Pfam sequence are then merged into the same file named after the Pfam number.

(3) Delete the files with a positive Pfam number. In the remaining Pfam number files, the negative data set (Liao et al., 2016) is extracted from the longest sequence of each Pfam.

(4) Use the CTDC method command to extract specific features in fasta files to generate GPCRs and non-GPCRs .csv files; positive GPCRs sample are marked as 0, negative sample are marked as −1, and the GPCRs and non-GPCRs .csv files are combined into one file.

(5) The combined .csv file was reduced by MRMD2.0, and the reduced CTDC-mRMD2.0.csv file was obtained.

(6) Select any two attributes of the 39 attributes in the CTDC sequence. GPCRs are purple and marked 0, and non-GPCRs

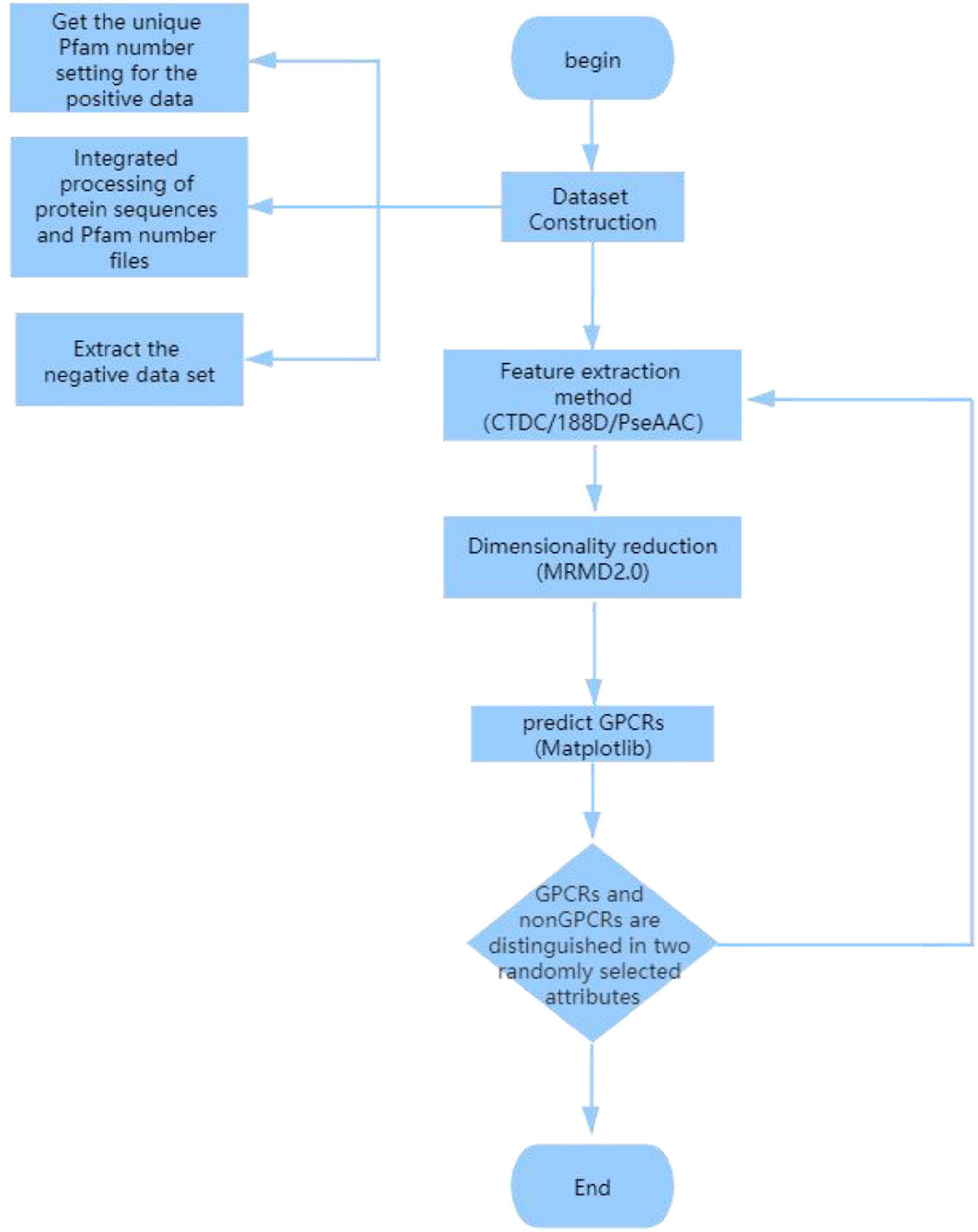

Get the unique Pfam number setting for the positive data
Integrated processing of protein sequences and Pfam number files
Extract the negative data set
begin
Dataset Construction
Feature extraction method (CTDC/188D/PseAAC)
Dimensionality reduction (MRMD2.0)
predict GPCRs (Matplotlib)
GPCRs and nonGPCRs are distinguished in two randomly selected attributes
End

FIGURE 3 | Experimental flow chart for prediction of G protein-coupled receptors.

are green and marked 1; Using Matplotlib, plot the picture of GPCRs and non-GPCRs.

RESULTS

Comparison of Effects of Different Features

CTDC was used to extract the characteristics of the GPCR protein feature sequences sample, including 39 properties. Previous studies showed that feature extraction is very important for constructing the computational predictors (Wei et al., 2017a,b; Xu et al., 2018b; Liang et al., 2019; Liu and Li, 2019; Patil and Chouhan, 2019; Shen et al., 2019; Zhang and Liu, 2019; Junwei et al., 2020; Liu et al., 2020; Wen et al., 2020). Any two of the 39 attributes were selected and plotted using Matplotlab to obtain the sample differentiation graph of GPCRs and non-GPCRs, as shown in **Figure 4**. Among them, the abscissa and the ordinate in the chart represent two of the 39 attributes. The x-coordinate of **Figure 4** on the left is the first of the 39 properties, "hydrophobicity_PRAM900101," named "RKEDQN," which is hydrophilic. The y-coordinate is the 14th property, "hydrophobicity_PRAM900101," named "GASTPHY," which is

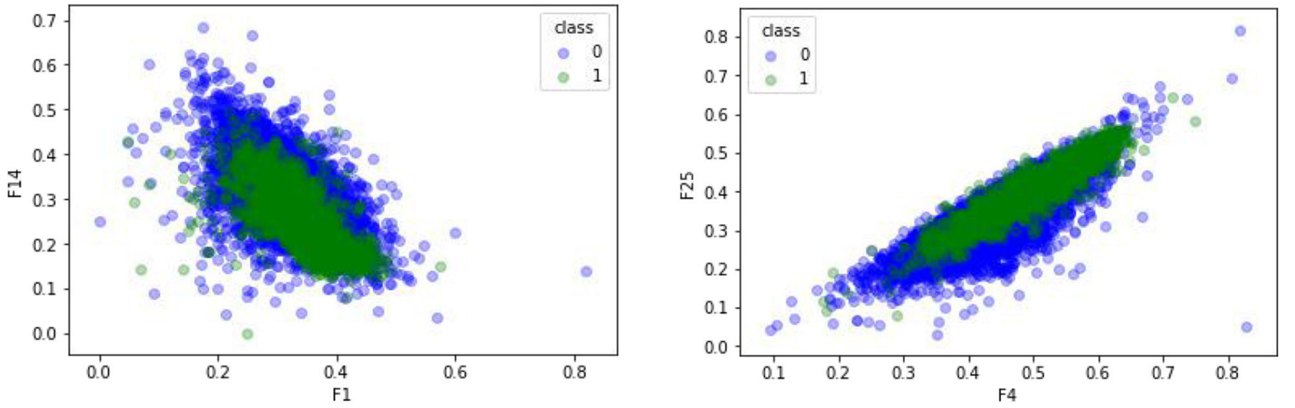


FIGURE 4 | Comparison of effects of different features.

neutral. In the right diagram of **Figure 4**, the X coordinate is the fourth attribute in the CTDC feature extraction method, normwaalsvolume: NVEQIL. The Y coordinate is the 25th attribute in CTDC, hydrophobicity_ENGD860101: CVLIMF. As seen from the chart, GPCRs and non-GRCRs are represented by blue and green, respectively, in which GPCRs and non-GPCRs can be clearly distinguished.

Comparison of Different Feature Extraction Methods

A comparative experiment was conducted, and the GPCR protein feature sequences are extracted by the 188D feature extraction method. The experimental effect is shown in **Figure 5**. In **Figure 5**, 120 and 100 dimensions of 188D are used. Non-GPCRs and GPCRs are marked as −1 and 1, respectively. It can be seen from the chart that the differentiation effect of GPCRs and non-GPCRs is very poor, but the differentiation effect of **Figure 4** is very good. Thus, whether GPCRs and non-GPCRs can be distinguished well is related to the selected feature extraction method.

Comparison of Results of Different Dimensionality Reduction Methods

The feature sequences of GPCR protein are extracted by the mRMR (Ding and Peng, 2005; Peng et al., 2005; Wang et al., 2018) dimensionality reduction method. 0 represents negative sample non-GPCRs, and 1 represents positive sample GPCRs. The experimental results are shown in **Figure 6**. In comparison with **Figure 4**, the two figures adopt the same feature extraction method of CTDC, the same attribute features and different dimension reduction methods. As seen from the figure, the difference between GPCRs and non-GPCRs was also very high after the dimension reduction method was used, and positive and negative samples are clearly distinguished.

Comparison With Others

In the study of Prediction of G Protein-Coupled Receptors with SVM-Prot Features and Random Forest (Liao et al., 2016), the researchers adopted a method different from the method in this paper to predict GPCRs and non-GPCRs. The experimental steps they adopted were as follows: 1. Extract GPCR and non-GPCR sample characteristics with 188D (Balfanz et al., 2013) 2. The sample sequences were divided into five parts, four of which were for the training set and the remaining one for the test set. In these four parts, positive and negative samples were treated with a strike balance 3. Random Forest was applied to the training samples, and the accuracy of the test samples was measured 4. Finally, Sn, Sp, Acc, MCC, and AUC standards were adopted to measure the accuracy. The correct classification rate of the five independent test sets was 90.64, 90.37, 88.04, 93.28, and 95.73, with an average rate of 91.61 ± 2.96%.

CONCLUSION

With the feature extraction method of CTDC, GPCRs and non-GPCRs can be well-distinguished from the two randomly selected dimensions. The same CTDC feature extraction method was used, but another dimension reduction method, mRMR, was selected. Compared with mRMD2.0, the differentiation effect was similar, and GPCRs and non-GPCRs could be significantly predicted. Using different feature extraction methods (188D) and the same dimensionality reduction method (mRMD2.0), GPCRs and non-GPCRs had no clear dividing line. In conclusion, different methods of feature extraction and the same method of dimensionality reduction have different effects on GPCRs and non-GPCRs. Therefore, the feature extraction method is the direct factor for distinguishing GPCRs from non-GPCRs.

However, a similar work was done in the Prediction of G protein-coupled sensor (Nordstrm et al., 2009) study. Compared with our study, the defects were as follows: 1. The 188D feature extraction method with more dimensions was adopted, the 188D feature extraction method had more feature dimensions, and the feature information of proteins was more complete and more comprehensive. The dimension information extracted by the CTDC method in this experiment has only 39 attribute characteristics, and there are less data. In addition, there is less redundant information after dimension reduction. 2. Five

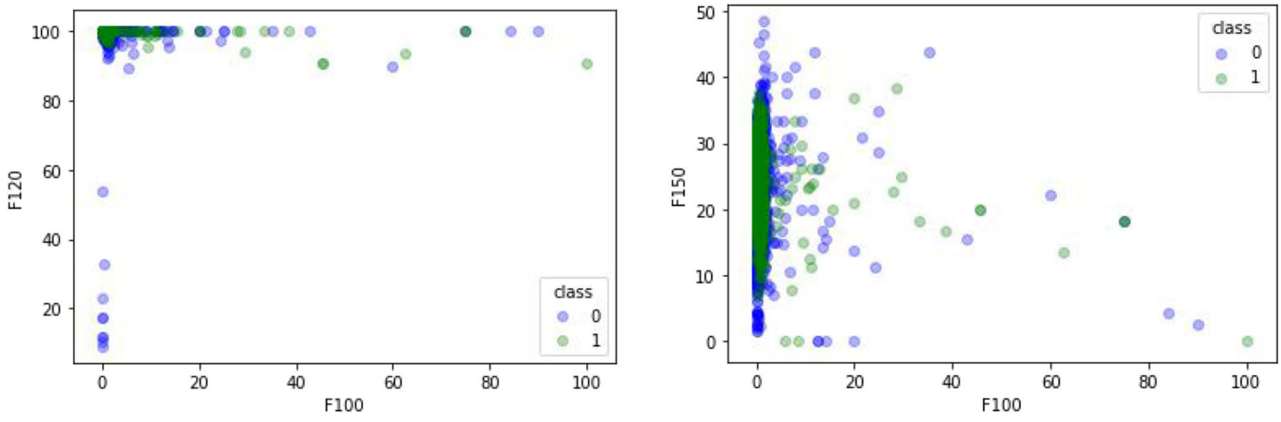


FIGURE 5 | Comparison of different feature extraction methods.

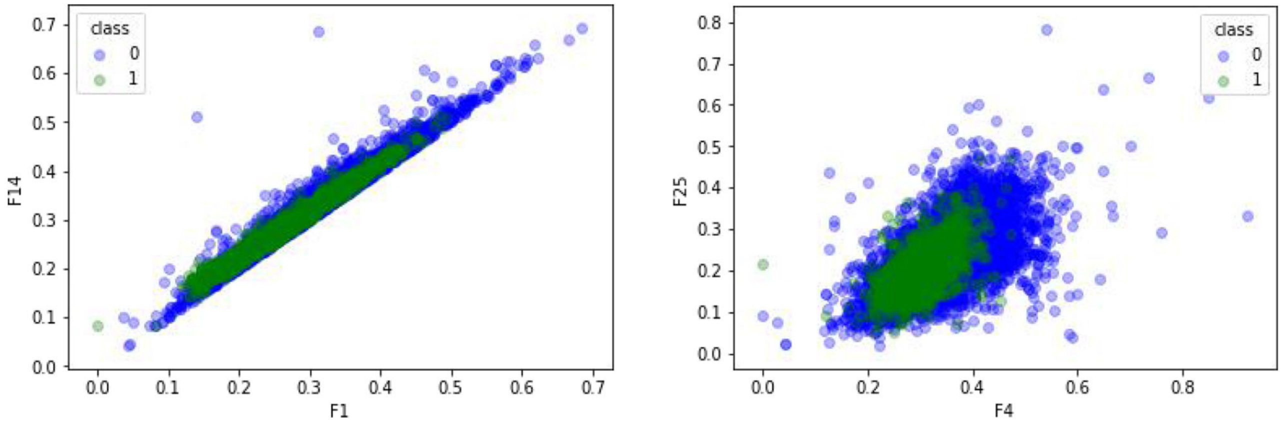


FIGURE 6 | Comparison of results of different dimensionality reduction methods.

independent test sets and training sets were divided in the Prediction of G protein-coupled sensor study, and the positive and negative samples in the training set tended to be balanced by the use of strike. However, defects in the strike method lead to inaccuracy of the data. In this paper, on the basis of original data collection, feature extraction and dimensionality reduction were directly carried out to distinguish GPCRs sample from non-GPCRs sample to obtain more accurate prediction results. Compared with this paper, the advantages are as follows: 1. The accuracy of the Prediction of G Protein by Coupled sensor study is approximately 90%; while the GPCRs and non-GPCRs differentiation diagram in this paper is shown by Matplotlab, and the accuracy was not calculated correctly. 2. The universality of this experiment is relatively low. The CTDC method and MRMD2.0 dimension reduction method may only be applicable to GPCRs protein sequence but not to other protein sequence. In the study of Prediction of G protein-coupled sensor, cross validation and Random Forest can be used on other protein sequences (Lai et al., 2018; Tang et al., 2018), especially the proposed framework can be applied to protein fold recognition (Wei et al., 2016; Liu et al., 2017), protein remote homology (Liu et al., 2020), protein subcellular localization (Lv et al., 2019), etc.

DISCUSSION

Like other macromolecules, proteins are important parts of the living body, the material basis of life, and they participate in almost every activity in the cell. Proteins perform many functions in the body. Through the study of proteins, the mechanism of diseases can be studied, and the design of new drugs can also be promoted. With the advent of machine learning, the function prediction of proteins has also flourished. Obtaining high-performance classification models, accurately and efficiently extracting protein sequences, and converting them into equal-length amino acid sequences have become research directions of many scientists.

Compared with the traditional experimental method, a set of experimental schemes in this paper replaces the redundant experimental steps. Using the CTDC method and dimensionality reduction in CTD, the redundant attributes in the protein sequence features are successfully removed, and they are drawn intuitively using Matplotlib. The division map between GPCRs and non-GPCRs is then drawn. In the division map, there can be a clear distinction between GPCRs and non-GPCRs. This experiment has achieved a certain degree of accuracy.

There are still many aspects that need to be further studied. The Matplotlib coordinate chart used to classify GPCRs and non-GPCRs can only distinguish the relatively large positive and negative samples after being divided by attributes, extracting several solutions: 1. The use of a single Matplotlib coordinate diagram is simple to operate and has many limitations; thus, it cannot reach high accuracy. In the later stage, more comprehensive computational intelligence method such as neural networks (Song et al., 2018a; Zhou et al., 2018; Bao et al., 2019; Hong et al., 2019; Sun et al., 2020), network methods (Sun et al., 2014; Zhou et al., 2015, 2016; Song et al., 2018b; Zeng et al., 2018) and evolutionary strategies (Xu et al., 2019a,b; Zeng et al., 2019b) can be adopted to take the extracted protein features as input. Thus, the positive and negative samples can be divided more accurately, and accuracy can be obtained. 2. In terms of high extraction accuracy, a more comprehensive protein feature extraction method combined with the dimension reduction method (Yang et al., 2019; Zhu et al., 2019) for GPCRs pruning was attempted to screen out features with higher differentiation between GPCRs and non-GPCRs.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/supplementary material.

AUTHOR CONTRIBUTIONS

ZC made the design of the subject and the whole idea of the whole experiment in the early stage. XG did comparative experiments and experimental data analysis. DW analyzed the results of the comparative experiment. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the Chinese National Natural Science Foundation under Grant 61876047.

REFERENCES

Balfanz, S., Jordan, N., Langenstück, T., Breuer, J., Bergmeier, V., and Baumann, A. (2013). Molecular, pharmacological, and signaling properties of octopamine receptors from honeybee (*Apis mellifera*) brain. *J. Neurochem*. 129, 284–296. doi: 10.1111/jnc.12619

Bao, S., Zhao, H., Yuan, J., Fan, D., Zhang, Z., Su, J., et al. (2019). Computational identification of mutator-derived lncRNA signatures of genome instability for improving the clinical outcome of cancers: a case study in breast cancer. *Brief. Bioinform.* bbz118. doi: 10.1093/bib/bbz118

Cai, C. Z., Han, L. Y., Ji, Z. L., Chen, X., and Chen, Y. Z. (2003). SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucl. Acids Res.* 31, 3692–3697. doi: 10.1093/nar/gkg600

Chen, W., Feng, P., Liu, T., and Jin, D. (2019). Recent advances in machine learning methods for predicting heat shock proteins. *Curr Drug Metab* 20, 224–228. doi: 10.2174/1389200219666181031105916

Chen, W., Nie, F., and Ding, H. (2020). Recent advances of computational methods for identifying bacteriophage virion proteins. *Protein Pept. Lett.* 27, 259–264. doi: 10.2174/0929866526666190410124642

Cheng, J. H., Yang, H., Liu, M. L., Su, W., Feng, P. M., Ding, H., et al. (2018a). Prediction of bacteriophage proteins located in the host cell using hybrid features. *Chemometr. Intell. Lab. Syst.* 180, 64–69. doi: 10.1016/j.chemolab.2018.07.006

Cheng, L. (2019). Computational and biological methods for gene therapy. *Curr. Gene Ther*. 19:210. doi: 10.2174/156652321904191022113307

Cheng, L., and Hu, Y. (2018). Human disease system biology. *Curr Gene Ther*. 18, 255–256. doi: 10.2174/1566523218666181010101114

Cheng, L., Jiang, Y., Ju, H., Sun, J., Peng, J., Zhou, M., et al. (2018b). InfAcrOnt: calculating cross-ontology term similarities using information flow by a random walk. *BMC Genom.* 19:919. doi: 10.1186/s12864-017-4338-6

Cheng, L., Zhao, H., Wang, P., Zhou, W., Luo, M., Li, T., et al. (2019). Computational methods for identifying similar diseases. Molecular therapy. *Nucl. Acids* 18, 590–604. doi: 10.1016/j.omtn.2019.09.019

Ding, C., and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *J. Bioinf. Comput. Biol.* 3, 185–205. doi: 10.1142/S0219720005001004

Ding, Y., Tang, J., and Guo, F. (2016). Predicting protein-protein interactions via multivariate mutual information of protein sequences. *Bmc Bioinformatics* 17:398. doi: 10.1186/s12859-016-1253-9

Ding, Y., Tang, J., and Guo, F. (2017). Identification of drug-target interactions via multiple information integration. *Inf. Sci.* 418, 546–560. doi: 10.1016/j.ins.2017.08.045

Ding, Y., Tang, J., and Guo, F. (2019). Identification of drug-side effect association via multiple information integration with centered kernel alignment. *Neurocomputing* 325, 211–224. doi: 10.1016/j.neucom.2018.10.028

Dubchak, I., Muchnik, I., Holbrook, S. R., and Kim, S. H. (1995). Prediction of protein folding class using global description of amino acid sequence. *Proc. Natl. Acad. Sci. U.S.A*. 92, 8700–8704. doi: 10.1073/pnas.92.19.8700

Dubchak, I., Muchnik, I., Mayor, C., and Dralyuk, I. (1999). Recognition of a protein fold in the context of the SCOP classification. *Proteins Struct. Funct. Bioinform.* 35, 401–407. doi: 10.1002/(SICI)1097-0134(19990601)35:4<401::AID-PROT3>3.0.CO;2-K

Feng, Y. M. (2019). Gene therapy on the road. *Curr. Gene Ther*. 19:6. doi: 10.2174/1566523219999190426144513

Fredriksson, R., Lagerström, M. C., Lundin, L. G., and Schiöth, H. B. (2003). The G-protein-coupled receptors in the human genome form five main families. Phylogenetic Analysis, Paralogon Groups, and Fingerprints. *Mol. Pharmacol.* 63, 1256–1272. doi: 10.1124/mol.63.6.1256

Guo, Y., Wu, C., Guo, M., Zou, Q., Liu, X., and Keinan, A. (2019). Combining sparse group lasso and linear mixed model improves power to detect genetic variants underlying quantitative traits. *Front. Genet*. 10:271. doi: 10.3389/fgene.2019.00271

Han, L. Y., Cai, C. Z., Ji, Z. L., Cao, Z. W., Cui, J., and Chen, Y. Z. (2004). Predicting functional family of novel enzymes irrespective of sequence similarity: a statistical learning approach. *Nucl. Acids Res.* 32, 6437–6444. doi: 10.1093/nar/gkh984

Hong, Z., Zeng, X., Wei, L., and Liu, X. J. B. (2019). Identifying enhancer-promoter interactions with neural network based on pre-trained DNA vectors and attention mechanism. *Bioinformatics* 36, 1037–1043. doi: 10.1093/bioinformatics/btz694

Islam, S. M. A., Heil, B. J., Kearney, C. M., and Baker, E. J. (2017). Protein classification using modified n-grams and skip-grams. *Bioinformatics*. 34, 1481–1487. doi: 10.1093/bioinformatics/btx823

Junwei, H., Xudong, H., Qingfei, K., and Liang, C. (2020). psSubpathway: a software package for flexible identification of phenotype-specific subpathways in cancer progression. *Bioinformatics.* 36, 2303–2305. doi: 10.1093/bioinformatics/btz894

Kentaro, T., and Minoru, K. (1996). Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins. *Protein Eng.* 9, 27–36.

Krishnan, A., Sällman, M., Robert, A., Helgi, F., and Schiöth, H. B. (2012). The origin of GPCRs: identification of mammalian like rhodopsin, adhesion, glutamate and frizzled GPCRs in fungi. *PLoS ONE* 7:e29817. doi: 10.1371/journal.pone.0029817

Lai, H. Y., Feng, C. Q., Zhang, Z. Y., Tang, H., Chen, W., and Lin, H. (2018). A brief survey of machine learning application in cancerlectin identification. *Curr. Gene Ther.* 18, 257–267. doi: 10.2174/1566523218666180913112751

Liang, C., Changlu, Q., He, Z., Tongze, F., and Xue, Z. (2019). gutMDisorder: a comprehensive database for dysbiosis of the gut microbiota in disorders and interventions. *Nucl. Acids Res.* 48, D554–D560. doi: 10.1093/nar/gkz843

Liao, Z., Ying, J., and Quan, Z. (2016). Prediction of G protein-coupled receptors with SVM-prot features and random forest. *Scientifica* 2016:8309253. doi: 10.1155/2016/8309253

Liu, B. (2019). BioSeq-analysis: a platform for DNA, RNA, and protein sequence analysis based on machine learning approaches. *Brief. Bioinf.* 20, 1280–1294. doi: 10.1093/bib/bbx165

Liu, B., Chen, S., Yan, K., and Weng, F. (2019b). iRO-PsekGCC: identify DNA replication origins based on Pseudo k-tuple GC composition. *Front. Genet.* 10:842. doi: 10.3389/fgene.2019.00842

Liu, B., Gao, X., and Zhang, H. (2019a). BioSeq-analysis2.0: an updated platform for analyzing DNA, RNA, and protein sequences at sequence level and residue level based on machine learning approaches. *Nucl. Acids Res.* 47:e127. doi: 10.1093/nar/gkz740

Liu, B., Jiang, S., and Zou, Q. (2020). HITS-PR-HHblits: protein remote homology detection by combining PageRank and hyperlink-induced topic search. *Brief. Bioinf.* 21, 298–308. doi: 10.1093/bib/bby104

Liu, B., Li, C., and Yan, K. (2019). DeepSVM-fold: protein fold recognition by combining Support Vector Machines and pairwise sequence similarity scores generated by deep learning networks. *Brief. Bioinf.* doi: 10.1093/bib/bbz098

Liu, B., and Li, K. (2019). iPromoter-2L2.0: identifying promoters and their types by combining Smoothing Cutting Window algorithm and sequence-based features. *Mol. Ther. Nucl. Acids* 18, 80–87. doi: 10.1016/j.omtn.2019.08.008

Liu, B., Luo, Z., and He, J. (2020). sgRNA-PSM: predict sgRNAs on-target activity based on position specific mismatch. *Mol. Ther. Nucl. Acids.* 20, 323–330. doi: 10.1016/j.omtn.2020.01.029

Liu, B., and Zhu, Y. (2019). ProtDec-LTR3.0: protein remote homology detection by incorporating profile-based features into Learning to Rank. *IEEE Access* 7, 102499–102507. doi: 10.1109/ACCESS.2019.2929363

Liu, B., Zhu, Y., and Yan, K. (2017). Fold-LTR-TCP: protein fold recognition based on triadic closure principle. *Brief. Bioinf.* 25. doi: 10.1093/bib/bbz139

Liu, K., and Chen, W. (2020). iMRM:a platform for simultaneously identifying multiple kinds of RNA modifications. *Bioinformatics.* 36, 3336–3342. doi: 10.1093/bioinformatics/btaa155

Lv, H., Dao, F. Y., Zhang, D., Guan, Z. X., Yang, H., Su, W., et al. (2020). iDNA-MS: an integrated computational tool for detecting DNA modification sites in multiple genomes. *iScience* 23:100991. doi: 10.1016/j.isci.2020.100991

Lv, Z., Jin, S., Ding, H., and Zou, Q. (2019). A random forest sub-Golgi protein classifier optimized via dipeptide and amino acid composition features. *Front. Bioeng. Biotechnol.* 7:215. doi: 10.3389/fbioe.2019.00215

Nordstrm, K. J. V., Linn, M. C. L., Wallér, M. J., Fredriksson, R., and Schith, H. B. (2009). The secretin GPCRs descended from the family of adhesion GPCRs. *Mol. Biol. Evol.* 26, 71–84. doi: 10.1093/molbev/msn228

Pang, H., Lin, A., Holford, M., Enerson, B., Lu, B., Lawton, M., et al. (2006). Pathway analysis using random forests classification and regression. *Bioinformatics* 22, 2028–2036. doi: 10.1093/bioinformatics/btl344

Patil, K., and Chouhan, U. (2019). Relevance of machine learning techniques and various protein features in protein fold classification: a review. *Curr. Bioinf.* 14, 688–697. doi: 10.2174/1574893614666190204154038

Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 1226–1238. doi: 10.1109/TPAMI.2005.159

Ru, X., Wang, L., Li, L., Ding, H., Ye, X., and Zou, Q. (2020). Exploration of the correlation between GPCRs and drugs based on a learning to rank algorithm. *Comput. Biol. Med.* 119:103660. doi: 10.1016/j.compbiomed.2020.103660

Shen, C., Jiang, L., Ding, Y., Tang, J., and Guo, F. (2019). LPI-KTASLP: prediction of lncRNA-protein interaction by semi-supervised link learning with multivariate information. *IEEE Access* 7, 13486–13496. doi: 10.1109/ACCESS.2019.2894225

Song, T., Rodríguez-Patón, A., Zheng, P., and Zeng, C. (2018a). Spiking neural P systems with colored spikes. *IEEE Trans. Cogn. Dev. Syst.* 10, 1106–1115. doi: 10.1109/TCDS.2017.2785332

Song, T., Zeng, X., Zheng, P., Jiang, M., and Rodríguez-Patón, A. (2018b). A parallel workflow pattern modelling using spiking neural p systems with colored spikes. *IEEE Trans. Nanobiosci.* 17, 474–484. doi: 10.1109/TNB.2018.2873221

Su, R., Wu, H., Xu, B., Liu, X., and Wei, L. (2019). Developing a multi-dose computational model for drug-induced hepatotoxicity prediction based on toxicogenomics data. *IEEE ACM Trans. Comput. Biol. Bioinf.* 16, 1231–1239. doi: 10.1109/TCBB.2018.2858756

Sun, J., Shi, H., Wang, Z., Zhang, C., Liu, L., Wang, L., et al. (2014). Inferring novel lncRNA-disease associations based on a random walk model of a lncRNA functional similarity network. *Mol. Biosyst.* 10, 2074–2081. doi: 10.1039/C3MB70608G

Sun, J., Zhang, Z., Bao, S., Yan, C., Hou, P., Wu, N., et al. (2020). Identification of tumor immune infiltration-associated lncRNAs for improving prognosis and immunotherapy response of patients with non-small cell lung cancer. *J. Immunother. Cancer* 8:e000110. doi: 10.1136/jitc-2019-000110

Tang, H., Zhao, Y. W., Zou, P., Zhang, C. M., Chen, R., Huang, P., et al. (2018). HBPred: a tool to identify growth hormone-binding proteins. *Int. J. Biol. Sci.* 14, 957–964. doi: 10.7150/ijbs.24174

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc.* 58, 267–288. doi: 10.1111/j.2517-6161.1996.tb02080.x

Wang, H., Ding, Y., Tang, J., and Guo, F. (2020). Identification of membrane protein types via multivariate information fusion with Hilbert-Schmidt Independence Criterion. *Neurocomputing* 383, 257–269. doi: 10.1016/j.neucom.2019.11.103

Wang, S. P., Zhang, Q., Lu, J., and Cai, Y. D. (2018). Analysis and prediction of nitrated tyrosine sites with the mRMR method and support vector machine algorithm. *Curr. Bioinform.* 13, 3–13. doi: 10.2174/1574893611666160608075753

Wei, L., Ding, Y., Su, R., Tang, J., and Zou, Q. (2018). Prediction of human protein subcellular localization using deep learning. *J. Parallel Distrib. Comput.* 117, 212–217. doi: 10.1016/j.jpdc.2017.08.009

Wei, L., Liao, M., Gao, X., and Zou, Q. (2015). An improved protein structural prediction method by incorporating both sequence and structure information. *IEEE Trans. Nanobiosci.* 14, 339–349. doi: 10.1109/TNB.2014.2352454

Wei, L., Wan, S., Guo, J., and Wong, K. K. (2017b). A novel hierarchical selective ensemble classifier with bioinformatics application. *Artif. Intell. Med.* 83, 82–90. doi: 10.1016/j.artmed.2017.02.005

Wei, L., Xing, P., Shi, G., Ji, Z., and Zou, Q. (2019). Fast prediction of methylation sites using sequence-based feature selection technique. *IEEE ACM Trans. Comput. Biol. Bioinf.* 16, 1264–1273. doi: 10.1109/TCBB.2017.2670558

Wei, L., Xing, P., Zeng, J., Chen, J., Su, R., and Guo, F. (2017a). Improved prediction of protein–protein interactions using novel negative samples, features, and an ensemble classifier. *Artif. Intell. Med.* 83, 67–74. doi: 10.1016/j.artmed.2017.03.001

Wei, L., Zou, Q., Liao, M., Lu, H., and Zhao, Y. (2016). A novel machine learning method for cytokine-receptor interaction prediction. *Combinatorial Chem. High Throughput Screen.* 19, 144–152. doi: 10.2174/1386207319666151110122621

Wen, S., Dong, M., Yang, Y., Zhou, P., Huang, T., and Chen, Y. (2020). End-to-end detection-segmentation network for face labeling. *IEEE Trans. Emerg. Top. Comput. Intell.* 1–11. doi: 10.1109/TETCI.2019.2947319

Xu, H., Zeng, W., Zeng, X., and Yen, G. G. (2019b). An evolutionary algorithm based on minkowski distance for many-objective optimization. *IEEE Trans. Cybernet.* 49, 3968–3979. doi: 10.1109/TCYB.2018.2856208

Xu, H., Zeng, W., Zhang, D., and Zeng, C. (2019a). MOEA/HD: a multiobjective evolutionary algorithm based on hierarchical decomposition. *IEEE Trans. Cybernet.* 49, 517–526. doi: 10.1109/TCYB.2017.2779450

Xu, L., Liang, G., Liao, C., Chen, G.-D., and Chang, C.-C. (2019c). k-skip-n-gram-RF: a random forest based method for Alzheimer's disease protein identification. *Front. Genet.* 10:33. doi: 10.3389/fgene.2019.00033

Xu, L., Liang, G., Shi, S., and Liao, C. (2018b). SeqSVM: a sequence-based support vector machine method for identifying antioxidant proteins. *Int. J. Mol. Sci.* 19:1773. doi: 10.3390/ijms19061773

Xu, L., Liang, G., Wang, L., and Liao, C. (2018a). A novel hybrid sequence-based model for identifying anticancer peptides. *Genes* 9:158. doi: 10.3390/genes9030158

Yan, K., Fang, X., Xu, Y., and Liu, B. (2019). Protein fold recognition based on multi-view modeling. *Bioinformatics* 35, 2982–2990. doi: 10.1093/bioinformatics/btz040

Yang, W., Zhu, X. J., Huang, J., Ding, H., and Lin, H. (2019). A brief survey of machine learning methods in protein sub-Golgi localization. *Curr. Bioinf.* 14, 234–240. doi: 10.2174/1574893613666181113131415

Zeng, X., Wang, W., Chen, C., and Yen, C. (2019b). A consensus community-based particle swarm optimization for dynamic community detection. *IEEE Trans. Cybernet.* 1–12. doi: 10.1109/TCYB.2019.2938895

Zeng, X., Zhu, S., Liu, X., Zhou, Y., Nussinov, R., and Cheng, F. (2019a). deepDR: a network-based deep learning approach to in silico drug repositioning. *Bioinformatics* 35, 5191–5198. doi: 10.1093/bioinformatics/btz418

Zeng, X., Zhu, S., Lu, W., Liu, Z., Huang, J., Zhou, Y., et al. (2020). Target identification among known drugs by deep learning from heterogeneous networks. *Chem. Sci.* 11, 1775–1797. doi: 10.1039/C9SC04336E

Zeng, X. X., Liu, L., Lu, L. Y., and Zou, Q. (2018). Prediction of potential disease-associated microRNAs using structural perturbation method. *Bioinformatics* 34, 2425–2432. doi: 10.1093/bioinformatics/bty112

Zhang, J., and Liu, B. (2019). A review on the recent developments of sequence-based protein feature extraction methods. *Curr. Bioinf.* 14, 190–199. doi: 10.2174/1574893614666181212102749

Zhang, X., Zou, Q., Rodriguez-Paton, A., and Zeng, X. (2019). and Bioinformatics, Meta-path methods for prioritizing candidate disease miRNAs. *IEEE ACM Trans. Comput. Biol. Bioinf.* 16, 283–291. doi: 10.1109/TCBB.2017.2776280

Zhang, Y., Liu, B., Dong, Q., and Jin, V. X. (2011). An improved profile-level domain linker propensity index for protein domain boundary prediction. *Protein Peptide Lett.* 18, 7–16. doi: 10.2174/092986611794328717

Zhou, M., Hu, L., Zhang, Z., Wu, N., Sun, J., and Su, J. (2018). Recurrence-associated long non-coding RNA signature for determining the risk of recurrence in patients with colon cancer. *Mol. Ther. Nucl. Acids* 12, 518–529. doi: 10.1016/j.omtn.2018.06.007

Zhou, M., Wang, X., Li, J., Hao, D., Wang, Z., Shi, H., et al. (2015). Prioritizing candidate disease-related long non-coding RNAs by walking on the heterogeneous lncRNA and disease network. *Mol. Biosyst.* 11, 760–769. doi: 10.1039/C4MB00511B

Zhou, M., Wang, X., Shi, H., Cheng, L., Wang, Z., Zhao, H., et al. (2016). Characterization of long non-coding RNA-associated ceRNA network to reveal potential prognostic lncRNA biomarkers in human ovarian cancer. *Oncotarget.* 7, 12598–12611. doi: 10.18632/oncotarget.7181

Zhu, X. J., Feng, C. Q., Lai, H. Y., Chen, W., and Lin, H. (2019). Predicting protein structural classes for low-similarity sequences by evaluating different features. *Knowl. Based Syst.* 163, 787–793. doi: 10.1016/j.knosys.2018.10.007

Zou, Q., Lin, G., Jiang, X., Liu, X., and Zeng, X. (2020). Sequence clustering in bioinformatics: an empirical study. *Brief. Bioinform.* 21, 1–10. doi: 10.1093/bib/bby090

Zou, Q., Wan, S., Ju, Y., Tang, J., and Zeng, X. (2016a). Pretata: predicting TATA binding proteins with novel features and dimensionality reduction strategy. *BMC Syst. Biol.* 10:114. doi: 10.1186/s12918-016-0353-5

Zou, Q., Wang, Z., Guan, X., Liu, B., Wu, Y., and Lin, Z. (2013). An approach for identifying cytokines based on a novel ensemble classifier. *BioMed Res. Int.* 2013:686090. doi: 10.1155/2013/686090

Zou, Q., Zeng, J., Cao, L., and Ji, R. (2016b). A novel features ranking metric with application to scalable visual and bioinformatics data classification. *Neurocomputing* 173, 346–354. doi: 10.1016/j.neucom.2014.12.123

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 23 July 2020
doi: 10.3389/fbioe.2020.00858

Identifying Antioxidant Proteins by Combining Multiple Methods

Xianhai Li[1,2]*, Qiang Tang*[2]*, Hua Tang*[2] *and Wei Chen*[1,2,3]*

[1] School of Pharmacy, Chengdu University of Traditional Chinese Medicine, Chengdu, China, [2] Innovative Institute of Chinese Medicine and Pharmacy, Chengdu University of Traditional Chinese Medicine, Chengdu, China, [3] School of Life Sciences, Center for Genomics and Computational Biology, North China University of Science and Technology, Tangshan, China

Antioxidant proteins play important roles in preventing free radical oxidation from damaging cells and DNA. They have become ideal candidates of disease prevention and treatment. Therefore, it is urgent to identify antioxidants from natural compounds. Since experimental methods are still cost ineffective, a series of computational methods have been proposed to identify antioxidant proteins. However, the performance of the current methods are still not satisfactory. In this study, a support vector machine based method, called Vote9, was proposed to identify antioxidants, in which the sequences were encoded by using the features generated from 9 optimal individual models. Results from jackknife test demonstrated that Vote9 is comparable with the best one of the existing predictors for this task. We hope that Vote9 will become a useful tool or at least can play a complementary role to the existing methods for identifying antioxidants.

Keywords: antioxidant, reduced amino acid composition, g-gap dipeptide composition, feature selection, support vector machine

Edited by:
Zhibin Lv,
University of Electronic Science and Technology of China, China

Reviewed by:
Shaherin Basith,
Ajou University, South Korea
Yongchun Zuo,
Inner Mongolia University, China

Correspondence:
Wei Chen
chenweiimu@gmail.com

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *30 May 2020*
Accepted: *03 July 2020*
Published: *23 July 2020*

Citation:
Li X, Tang Q, Tang H and Chen W (2020) Identifying Antioxidant Proteins by Combining Multiple Methods. Front. Bioeng. Biotechnol. 8:858. doi: 10.3389/fbioe.2020.00858

INTRODUCTION

Reactive oxygen species (ROS) are composed of oxygen free radicals and nitrogen free radicals. Free radicals contain unpaired electron molecules or atoms, which are generally unstable and highly reactive. They can trigger lipid peroxidation during metabolism, which leads to DNA strand breaks, and even oxidize biofilms and almost all molecules in tissues indiscriminately. Fortunately, organisms have evolved effective strategies to detect and prevent molecular oxygen metabolites (Finkel and Holbrook, 2000; Mccord, 2000; Klaus and Heribert, 2004; Li et al., 2015). This is called the antioxidant system of organisms, which can effectively resist the damages caused by ROS (Agus et al., 2011).

Owing to their important roles in the antioxidant system, natural antioxidants have received more and more attentions (Yigit et al., 2014). Antioxidant proteins can neutralize free radicals, thereby blocking cell damage or death caused by free radicals. The consumption of antioxidants can be used to reduce the oxidative stress caused by excessive ROS, and reduce the damage to the organism (Yang et al., 2017). Antioxidants have also been applied to prevent diseases such as heart disease, cancer, cardiovascular disease (Gey, 1990; Dreher and Junod, 1996; Diaz et al., 1997). Its unique role in anti-aging was also reported (Ames et al., 1993).

Accordingly, many proteins extracted from rapeseed, ginkgo and other plant seeds are used as natural antioxidants (Nichole et al., 2008; Huang et al., 2009). Some micronutrients such as vitamin C and vitamin E (Lobo et al., 2010) are also considered as antioxidant molecules. However, our body cannot synthesize these nutrients, so we need to ingest them from the diet. Therefore, it has become an urgent task to identify proteins with antioxidant activity from natural compounds.

Although identifying antioxidant proteins through biochemical experiments is an objective and accurate method, they are still labor intensive and expensive. With the massive production of protein sequences, a series of computational methods have been proposed to identify antioxidant proteins. For the first time, Enrique et al. (2013) proposed a random forest model for predicting antioxidant proteins based on star map topological index and achieved satisfactory results. However, their model was trained based on a dataset including redundant sequences that might lead to overestimation problems (Chou, 2011). In 2013, Feng et al. (2013) constructed a high quality dataset with the sequence similarity less than 60%. Based on this dataset, they developed a Naive Bayes method by using the optimal dipeptides and obtained an average accuracy of 66.88%. Based on this dataset, a series of methods have been proposed in recent years. In 2016, Feng et al. (2016) proposed a support vector machine based method, called AodPred, which identifies antioxidant by using the optimal 3-gap dipeptide features and improves the prediction accuracy to 74.79%. Later on, Lei et al. (2018) developed a computational model called SeqSVM by using support vector machine and obtained an overall accuracy of 89.46%. More recently, Meng et al. (2019) proposed another support vector machine model called AOPs-SVM by integrating multiple kinds of features and obtained an overall accuracy of 94.2%. However, the sensitivity of AOPs-SVM is only 68%.

The above results indicate that the prediction accuracy still needs to be improved. Therefore, in this study, based on the optimal dipeptide composition and the reduced amino acid composition (Chen D. et al., 2012; Chen W. et al., 2012; Feng et al., 2016; Lv et al., 2019), a new model was constructed. The results show that the performance of the proposed method for identifying antioxidant proteins is better than or at least comparable to existing methods.

MATERIALS AND METHODS

Training Set and Test Set

The dataset used in the present work is the same as the one used by Feng et al. (2013, 2017),which includes 253 antioxidant protein sequences and 1552 non-antioxidant protein sequences with the sequence identity less than 60%. The dataset is expressed as:

$$S = S_{+} \cup S_{-} \tag{1}$$

where "S" stands for benchmark dataset, "S_{+}" is the positive dataset and contains 253 antioxidant protein sequences, and "S_{-}" is the negative dataset and contains 1552 non-antioxidant protein sequences. The longest and shortest peptides in the dataset are 1463 and 11 amino acids, respectively.

In the following analysis, the dataset S was divided into two parts. One of them is the training set S_T and includes 80% of the sequences in S, and the remaining 20% sequences form the testing set S_E, which are expressed as following,

$$S_T = S_{+}^{*}0.8 \cup S_{-}^{*}0.8 \tag{2}$$

$$S_E = S - S_T \tag{3}$$

Independent Dataset

To objectively evaluate the proposed method and compare with its counterpart, an independent dataset was built in the present work. By searching the Universal Protein Resource (Uniprot) with the keywords "antioxidant" and "reviewed," and setting the date from March 1, 2014 to March 31, 2020, we obtained 22 antioxidant protein sequences that are independent from the sequences in the dataset S.

Support Vector Machine

Support Vector Machine (SVM) is a method for effectively identifying data according to supervised learning method, which is widely used in bioinformatics and other fields (Feng et al., 2016; Liao et al., 2018; Wang et al., 2019; Liu and Chen, 2020). If the samples are linearly separated, the basic idea of the SVM algorithm is to solve the separation hyperplane that can correctly divide the training dataset and have the largest geometric interval; when the samples are nonlinearly separated, SVM maps the low-dimensional data to the high-dimensional data by the kernel function space. In this work, the LIBSVM package downloaded from https://www.csie.ntu.edu.tw/~cjlin/libsvm/ was used to perform the prediction. The best regularization parameter *C* and kernel width parameter *g* were determined by using the grid search method.

Sequence Representation

g-gap Dipeptide Composition

The *g*-gap dipeptide composition was proposed to describe the long-range correlation between two amino acid residues and has been proved to be effective in the field of protein recognition (Ding et al., 2013; Lin et al., 2013; Tan et al., 2019). Accordingly, in the present work, the *g*-gap dipeptide composition was used to encode the sequences in both benchmark dataset and independent test dataset.

The g-gap dipeptide composition is expressed as following,

$$F = [f_1^g \, f_2^g \ldots f_i^g \ldots f_{400}^g]^T \tag{4}$$

$$f_i^g = \frac{n_i^g}{L - g - 1} \tag{5}$$

where $f_{\mathrm{i}}^{\mathrm{g}}$ represents the frequency of the *i*-th (*i* = 1, 2,..., 400) dipeptide with *g*-gap interval in the protein sequence, and T

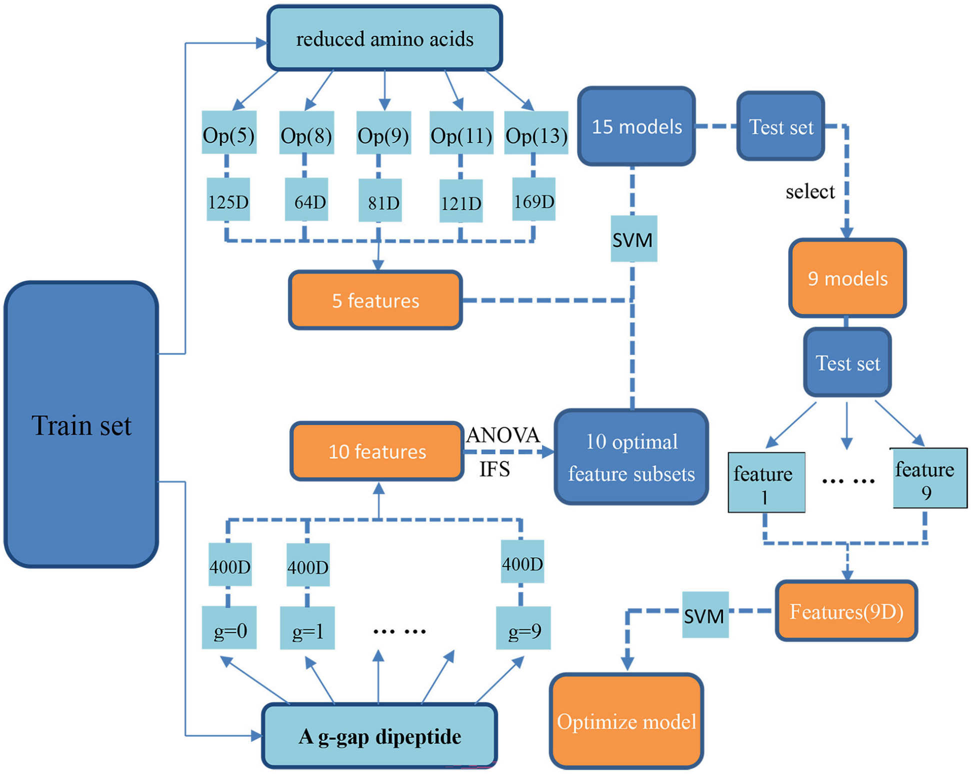
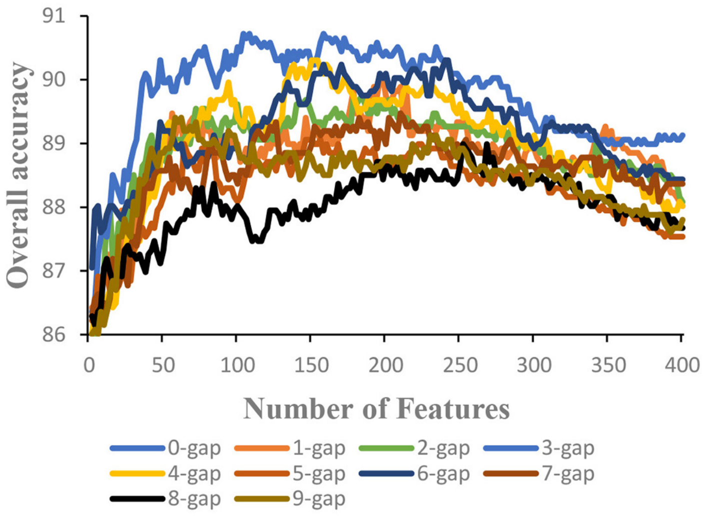


FIGURE 1 | The flowchart of building the proposed method. The samples in the training dataset were firstly encoded by using reduced amino acid compositions and the optimal g-gap dipeptide compositions, respectively. Accordingly, 15 SVM models based on these different kinds of features was built. After validating the combinational performance of these models on the test dataset, 9 of the 15 SVM models were selected out as the optimal models. Finally, the SVM outs of these 9 models were used as the new features and used as the inputs of the SVM for building the proposed model.



g-gap	optimal features	accuracy
0-gap	172	0.886
1-gap	197	0.898
2-gap	184	0.892
3-gap	158	0.909
4-gap	150	0.903
5-gap	147	0.9
6-gap	158	0.9
7-gap	210	0.898
8-gap	252	0.875
9-gap	58	0.873

FIGURE 2 | The IFS curves of different g-gap dipeptides (g = 0, 1, 2,..., 9). The optimal number of features and the accuracy based on the optimal features were shown in the right of the figure.

represents the transposition of the vector. n_i^g represents the number of the *i*-th *g*-gap dipeptide. In the present work, *g* is an integer in the range of [0, 9]. For example, *g* = 0 represents the correlation between two adjacent amino acid residues, and *g* = 1 represents the correlation of two amino acid residues separated by one residue, and so forth.

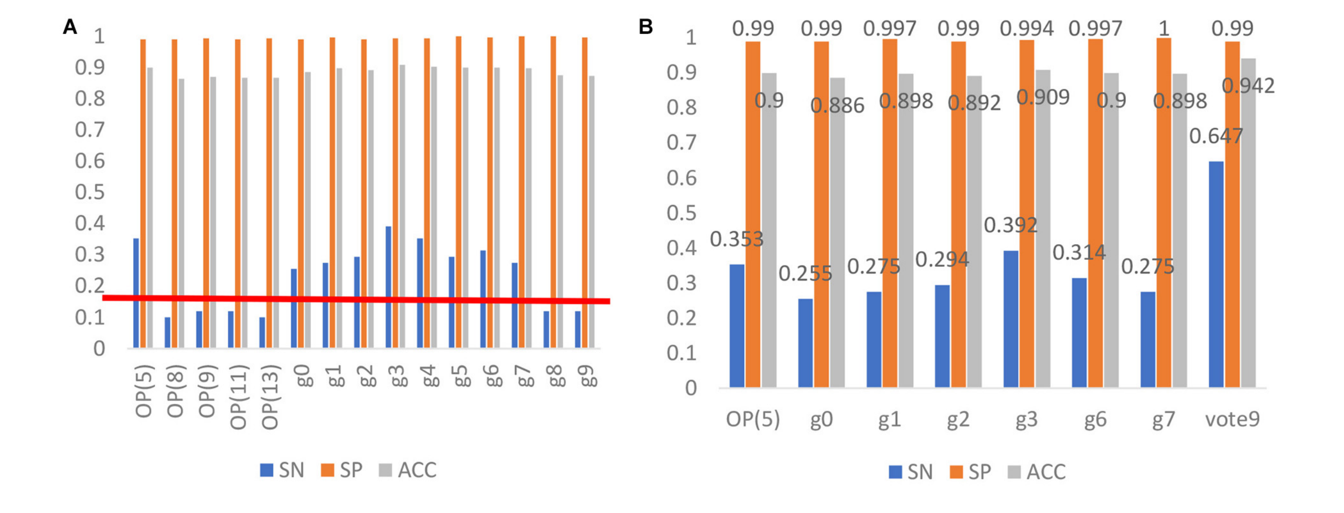


FIGURE 3 | (A) The performance of the 15 models for identifying antioxidants. OP (5) stands for the method of optimizing amino acid residues to divide 20 amino acid letters into 5 categories, and then uses LIBSVM to establish a classification model. Using the method of g-gap dipeptide (Feng et al., 2016), we selected the best feature subset of protein sequence steps g = 0, 1... 7 to construct a g0, g1... g7 classification model. Vote9 is a comprehensive classification model that used the prediction results of the above classification models as feature vectors. **(B)** Comparison between Vote9 and single classification model.

Reduced Amino Acid Composition

With the aim of including structural information, the reduced amino acid composition (RAAC) was applied to encode proteins (Feng et al., 2016). Compared with the classical amino acid composition, the RAACs can reduce protein complexity and eliminate part of the redundant signals without losing sequence information intact (Wang and Wang, 1999; Liu et al., 2018). In order to obtain the RAAC from the sequences, Zuo et al. (2017) established the online webserver and database (Zheng et al., 2019) that can be used to calculate RAAC.

In term of RAAC, based on amino acid sequence and structure information, the 20 natural amino acids can be aggregated into a smaller number of representative amino acid residues (Thomas and Dill, 1996; Mirny and Shakhnovich, 1999; Solis and Rackovsky, 2000). According to the different optimization procedures (Op) for protein sequences proposed by Etchebest et al. (2007), there are 5 different cluster files for the 20 natural amino acids, i.e., Op(5), Op(8), Op(9), Op(11)and Op(13), which are formulated as below:

$$\mathrm{Op}\ (i) =$$

$$\begin{cases} \mathrm{Op}\ (5) : \{\mathrm{G};\ \mathrm{IVFYW};\ \mathrm{ALMEQRK};\ \mathrm{P};\ \mathrm{NDHSTC}\} \\ \mathrm{Op}\ (8) : \{\mathrm{G};\ \mathrm{IV};\ \mathrm{FYW};\ \mathrm{ALM};\ \mathrm{EQRK};\ \mathrm{P};\ \mathrm{ND};\ \mathrm{HSTC}\} \\ \mathrm{Op}\ (9) : \{\mathrm{G};\ \mathrm{IV};\ \mathrm{FYW};\ \mathrm{ALM};\ \mathrm{EQRK};\ \mathrm{P};\ \mathrm{ND};\ \mathrm{HS};\ \mathrm{TC}\} \\ \mathrm{Op}\ (11) : \{\mathrm{G};\ \mathrm{IV};\ \mathrm{FYW};\ \mathrm{A};\ \mathrm{LM};\ \mathrm{EQRK};\ \mathrm{P};\ \mathrm{ND};\ \mathrm{HS};\} \\ \mathrm{T};\ \mathrm{C}\} \\ \mathrm{Op}\ (13) : \{\mathrm{G};\ \mathrm{IV};\ \mathrm{FYW};\ \mathrm{A};\ \mathrm{L};\ \mathrm{M};\ \mathrm{E};\ \mathrm{QRK};\ \mathrm{P};\ \mathrm{ND}; \\ \ \mathrm{HS};\ \mathrm{T};\ \mathrm{C}\} \end{cases} \quad (6)$$

where i indicates the different cluster profiles (i = 5, 8, 9, 11, 13), and the letters between the two semicolons belong to the same cluster.

Accordingly, a sequence can be encoded based on the reduced amino acid composition. As indicated in Eq. 6, for the n-peptide composition with various cluster profiles, the components and dimensions of the feature vector will be different.

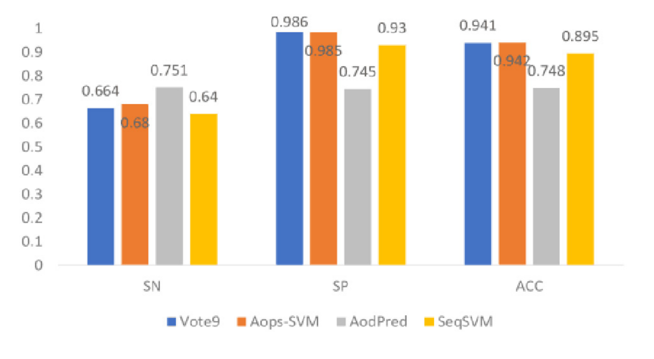


FIGURE 4 | Comparison of Vote9 with existing methods.

$$\Psi = [\Psi_1,\ \Psi_2,\ \cdots,\ \Psi_\Omega]^T \quad (7)$$

where Ω is the dimension of the vector, and is based on the selected n and cluster profiles. For example, for the dipeptide composition with the cluster profile of Op(5), the Ω will be 25. In the current work, our initial tests demonstrate that the optimal n for different cluster profiles is as following,

$$\Omega = \begin{cases} 5^{3\,=}125 & \textit{for Op}(5)\ \textit{cluster} \\ 8^{2\,=}64 & \textit{for Op}(8)\ \textit{cluster} \\ 9^{2\,=}81 & \textit{for Op}(9)\ \textit{cluster} \\ 11^{2\,=}121 & \textit{for Op}(11)\ \textit{cluster} \\ 13^{2\,=}169 & \textit{for Op}(13)\ \textit{cluster} \end{cases} \quad (8)$$

Performance Evaluation

There are usually three methods for evaluating the performance of computational models, namely independent dataset test, k-fold cross-validation test, and jackknife test (Wei et al., 2017; Chen et al., 2019; Manavalan et al., 2019a,b; Yang et al., 2019; Hasan et al., 2020; Lv et al., 2020). Among the three evaluation methods, the most rigorous and least random jackknife test was used to evaluate the proposed method.

The sensitivity (Sn), specificity (Sp), accuracy (Acc) and Mathew's correlation coefficient (MCC) was selected as the evaluation metrics that are defined as following,

$$Sn = \frac{TP}{TP + FN} \tag{9}$$

$$Sp = \frac{TN}{TN + FP} \tag{10}$$

$$Acc = \frac{TP + TN}{TP + FN + TN + FP} \tag{11}$$

$$MCC = \frac{TN^{*}TP - FP^{*}FN}{\sqrt{(TP + FP)^{*}(FN + TN)^{*}(TP + FN)^{*}(TN + FP)}} \tag{12}$$

where TP, FP, FN, and TN represent true positive, false positive, false negative and true negative, respectively.

Feature Selection

The principle of analysis of variance (ANOVA) is to measure the characteristic variance by calculating the ratio (*F*-value) between the characteristics of the groups and the internal characteristics of the groups (Lin and Ding, 2011; Basith et al., 2019). The larger the *F*-value, the greater the probability that each sample comes from a different population. In order to exclude redundant features and enhance the robustness of the proposed model, the ANOVA that widely used in computational proteomics (Ding et al., 2013; Lin et al., 2013; Basith et al., 2020) combined with the incremental feature selection (IFS) strategy was used to select the optimal features.

Flowchart of the Method

By following the above procedure, we proposed a new computational method for identifying antioxidants. The flowchart of how to build it was shown in **Figure 1**.

RESULTS AND DISCUSSION

Prediction Performance

In order to obtain the optimal features, for a given kind of g-gap dipeptide composition, the 400 g-gap dipeptide compositions were ranked based on their F-scores. Each of the 400 dipeptide compositions were added one by one from higher to lower rank. This procedure was repeated 400 times, and for each time a SVM model was built. The accuracies of these models were then used to plot the IFS curve. Accordingly, the 10 IFS curves for $g = 0$ to 9 were obtained (**Figure 2**), where the abscissa is the number of features and the ordinate is the corresponding accuracy. In each curve, the optimal number of features were obtained when the curve reaches its peak. The optimal number of features and the accuracy based on the optimal features were shown in the right of **Figure 2**. Accordingly, 10 models were obtained based on g-gap dipeptide compositions.

Based on the reduced amino acid composition, another five models were built for identifying antioxidants. Their predictive performances together with that of the 10 models based on g-gap dipeptide composition were indicated in **Figure 3A**.

According to the prediction results of the 15 models, we removed 6 models with the sensitivity less than 20%. Therefore, 9 models were left and were combined to build the final model in the following analysis. To do so, the out of the nine SVM based models (1 or −1) were further used as the input of the SVM. Therefore, each sequence will be re-encoded by a 9-dimension vector with the element of 1 or −1. The model thus obtained is called Vote9. In the jackknife test, Vote9 obtained an accuracy of 0.94 with the sensitivity of 0.65, specificity of 0.99 and MCC of 0.74.

Comparison With Single Model

In order to demonstrate the better performance of Vote9, we compared its performance with that of the single model for identifying antioxidants in the test dataset. The result is shown in **Figure 3B**. It was found that the sensitivity, specificity and accuracy of Vote9 are all significantly better than those of any

TABLE 1 | Comparative results of different methods for identifying antioxidants in independent dataset.

Sample	Aops-SVM	Aodpred	Vote9	Sample	Aops-SVM	Aodpred	Vote9
P9WQB7	Y	Y	N	P9WIS6	Y	N	N
P9WHH9	Y	N	N	P9WQB6	Y	Y	N
P9WIS7	Y	N	**Y**	P9WID9	Y	Y	N
P9WG35	Y	Y	N	O17433	Y	Y	N
P9WGE9	Y	Y	N	P9WIE0	Y	N	N
P9WQB5	Y	Y	N	P9WID8	Y	Y	N
P9WIE3	Y	Y	N	P9WGE8	Y	Y	N
P0CU34	Y	Y	N	C0HK70	Y	Y	N
Q5ACV9	N	N	N	P9WQB4	Y	Y	N
P9WHH8	Y	N	**Y**	P9WG34	Y	Y	N
P9WIE1	Y	N	**Y**	P9WIE2	Y	Y	N

single model, demonstrating that it's necessary to built the model by combining the optimal single models.

Comparison With Existing Methods

In this section, we compared the performance of Vote9 with the performance of other existing methods (Aops-SVM, AodPred, and SeqSVM) that all trained based on the same dataset. Their performances were shown in **Figure 4**.

It was found that the accuracy of Vote9 is better than that of AodPred and SeqSVM, and is comparable with that of Aops-SVM. Although the sensitivity of Vote9 is lower than that of Aops-SVM and AodPred, its specificity is higher than that of the other three methods (Aops-SVM, AodPred, and SeqSVM). This result indicate that Vote9 might also become a useful tool for identifying antioxidants.

In order to objectively evaluate the performance of different methods for identifying antioxidants, a comparison was performed based on the independent dataset. Since some of the previous methods didn't provide publicly available tool or doesn't work properly, the comparison was also performed among Vote9, Aops-SVM, and AodPred. Their performances for identifying antioxidants in independent dataset were reported in **Table 1**. As shown in **Table 1**, we found that Aops-SVM performs the best, and Vote9 and AodPred can be used as complementary tools.

Conclusion

The role of antioxidant proteins in neutralizing free radicals and preventing the damage of free radicals to cells is well known. Unfortunately, there are very few molecules with antioxidant properties in nature. Therefore, in order to accelerate researches on antioxidant proteins, there is an urgent need to develop effective methods for identifying them.

In the present work, we proposed a new method, called Vote9, in which the sequences were encoded by using the features generated from 9 optimal individual models. Results from jackknife test demonstrated that Vote9 is comparable with the best of the existing predictors for this task. The results of independent dataset test demonstrate that Vote9 can play a complementary role to the existing methods in this area. We hope that Vote9 will become a useful method for identifying antioxidants.

DATA AVAILABILITY STATEMENT

All datasets presented in this study are included in the article/supplementary material.

AUTHOR CONTRIBUTIONS

WC conceived and designed the experiments. XL, QT, and HT performed the experiments. XL and WC wrote the manuscript. All authors read and approved the final manuscript.

REFERENCES

Agus, S. T., Eka, M., Oh, L. K., and Keizo, H. (2011). Isolation and characterization of antioxidant protein fractions from melinjo (*Gnetum gnemon*) seeds. *J. Agric. Food Chem.* 59, 5648–5656. doi: 10.1021/jf2000647

Ames, B. N., Shigenaga, M. K., and Hagen, T. M. (1993). Oxidants, antioxidants, and the degenerative diseases of aging. *Proc. Natl. Acad. Sci. U.S.A.* 90, 7915–7922. doi: 10.1073/pnas.90.17.7915

Basith, S., Manavalan, B., Hwan Shin, T., and Lee, G. (2020). Machine intelligence in peptide therapeutics: a next-generation tool for rapid disease screening. *Med. Res. Rev.* doi: 10.1002/med.21658 [Epub ahead of print].

Basith, S., Manavalan, B., Shin, T. H., and Lee, G. (2019). SDM6A: a web-based integrative machine-learning framework for predicting 6mA sites in the rice genome. *Mol. Ther. Nucleic Acids* 18, 131–141. doi: 10.1016/j.omtn.2019.08.011

Chen, D., Lu-Feng, Y., Shou-Hui, G., Hao, L., and Wei, C. (2012). Identification of mycobacterial membrane proteins and their types using over-represented tripeptide compositions. *J. Proteom.* 77, 321–328. doi: 10.1016/j.jprot.2012.09.006

Chen, W., Feng, P., and Lin, H. (2012). Prediction of replication origins by calculating DNA structural properties. *FEBS Lett.* 586, 934–938. doi: 10.1016/j.febslet.2012.02.034

Chen, W., Feng, P., Liu, T., and Jin, D. (2019). Recent advances in machine learning methods for predicting heat shock proteins. *Curr. Drug Metab.* 20, 224–228. doi: 10.2174/1389200219666181031105916

Chou, K.-C. (2011). Some remarks on protein attribute prediction and pseudo amino acid composition. *J. Theor. Biol.* 273, 236–247. doi: 10.1016/j.jtbi.2010.12.024

Diaz, M. N., Frei, B., Vita, J. A., and Keaney, J. F. (1997). Antioxidants and atherosclerotic heart disease. *N. Engl. J. Med.* 337, 408–416. doi: 10.1056/nejm199708073370607

Ding, H., Guo, S.-H., Deng, E.-Z., Yuan, L.-F., Guo, F.-B., Huang, J., et al. (2013). Prediction of Golgi-resident protein types by using feature selection technique. *Chemometr. Intellig. Lab. Syst.* 124, 9–13. doi: 10.1016/j.chemolab.2013.03.005

Dreher, D., and Junod, A. F. (1996). Role of oxygen free radicals in cancer development. *Eur. J. Cancer* 32, 30–38. doi: 10.1016/0959-8049(95)00531-5

Enrique, F.-B., Vanessa, A.-P., Robert, M. C., and Julian, D. (2013). Random forest classification based on star graph topological indices for antioxidant proteins. *J. Theor. Biol.* 317, 331–337. doi: 10.1016/j.jtbi.2012.10.006

Etchebest, C., Benros, C., Bornot, A., Camproux, A.-C., and Brevern, A. G. (2007). A reduced amino acid alphabet for understanding and designing protein adaptation to mutation. *Eur. Biophys. J.* 36, 1059–1069. doi: 10.1007/s00249-007-0188-5

Feng, P., Ding, H., Lin, H., and Chen, W. (2017). AOD: the antioxidant protein database. *Sci. Rep.* 7:7449.

Feng, P., Chen, W., and Lin, H. (2016). Identifying antioxidant proteins by using optimal dipeptide compositions. *Interdiscipl. Sci. Comput. Life Sci.* 8, 186–191. doi: 10.1007/s12539-015-0124-9

Feng, P.-M., Hao, L., and Wei, C. (2013). Identification of antioxidants from sequence information using naïve Bayes. *Comput. Math. Mat. Methods Med.* 2013:567529.

Finkel, T., and Holbrook, N. J. (2000). Oxidants, oxidative stress and the biology of ageing. *Nature* 408, 239–247. doi: 10.1038/35041687

Gey, K. F. (1990). The antioxidant hypothesis of cardiovascular disease: epidemiology and mechanisms. *Biochem. Soc. Trans.* 18, 1041–1045. doi: 10.1042/bst0181041

Hasan, M. M., Schaduangrat, N., Basith, S., Lee, G., Shoombuatong, W., and Manavalan, B. (2020). HLPpred-Fuse: improved and robust prediction of hemolytic peptide and its activity by fusing multiple feature representation. *Bioinformatics* 36, 3350–3356. doi: 10.1093/bioinformatics/btaa160

Huang, W., Deng, Q., Xie, B., Shi, J., Huang, F., Tian, B., et al. (2009). Purification and characterization of an antioxidant protein from *Ginkgo biloba* seeds. *Food Res. Intern.* 43, 86–94. doi: 10.1016/j.foodres.2009.08.015

Klaus, A., and Heribert, H. (2004). Reactive oxygen species: metabolism, oxidative stress, and signal transduction. *Annu. Rev. Plant Biol.* 55, 373–399. doi: 10.1146/annurev.arplant.55.031903.141701

Lei, X., Guangmin, L., Shuhua, S., and Changrui, L. (2018). SeqSVM: a sequence-based support vector machine method for identifying antioxidant proteins. *Intern. J. Mol. Sci.* 19:1773. doi: 10.3390/ijms19061773

Li, S., Tan, H.-Y., Wang, N., Zhang, Z.-J., Lao, L., Wong, C.-W., et al. (2015). The role of oxidative stress and antioxidants in liver diseases. *Intern. J. Mol. Sci.* 16, 26087–26124.

Liao, Z. J., Li, D. P., Wang, X. R., Li, L. S., and Zou, Q. (2018). Cancer diagnosis through isomir expression with machine learning method. *Curr. Bioinform.* 13, 57–63. doi: 10.2174/1574893611666160609081155

Lin, H., Chen, W., and Ding, H. (2013). AcalPred: a sequence-based tool for discriminating between acidic and alkaline enzymes. *PLoS One* 8:75726. doi: 10.1371/journal.pone.0075726

Lin, H., and Ding, H. (2011). Predicting ion channels and their types by the dipeptide mode of pseudo amino acid composition. *J. Theor. Biol.* 269, 64–69. doi: 10.1016/j.jtbi.2010.10.019

Liu, D., Li, G., and Zuo, Y. (2018). Function determinants of TET proteins: the arrangements of sequence motifs with specific codes. *Brief. Bioinform.* 20, 1826–1835.

Liu, K., and Chen, W. (2020). iMRM:a platform for simultaneously identifying multiple kinds of RNA modifications. *Bioinformatics* 36, 3336–3342. doi: 10.1093/bioinformatics/btaa155

Lobo, V., Patil, A., Phatak, A., and Chandra, N. (2010). Free radicals, antioxidants and functional foods: impact on human health. *Pharm. Rev.* 4, 118–126.

Lv, H., Dao, F., Zhang, D., Guan, Z., Yang, H., Su, W., et al. (2020). iDNA-MS: an integrated computational tool for detecting DNA modification sites in multiple genomes. *Science* 23:100991. doi: 10.1016/j.isci.2020.100991

Lv, Z., Jin, S., Ding, H., and Zou, Q. (2019). A random forest sub-Golgi protein classifier optimized via dipeptide and amino acid composition features. *Front. Bioeng. Biotechnol.* 7:215. doi: 10.3389/fbioe.2019.00215

Manavalan, B., Basith, S., Shin, T. H., Wei, L., and Lee, G. (2019a). Meta-4mCpred: a sequence-based meta-predictor for accurate DNA 4mC site prediction using effective feature representation. *Mol. Ther. Nucleic Acids* 16, 733–744. doi: 10.1016/j.omtn.2019.04.019

Manavalan, B., Shaherin, B., Hwan, S. T., Yeon, L. D., Leyi, W., and Gwang, L. (2019b). 4mCpred-EL: an ensemble learning framework for identification of DNA N4-methylcytosine sites in the mouse genome. *Cells* 8:1332. doi: 10.3390/cells8111332

Mccord, J. M. (2000). The evolution of free radicals and oxidative stress. *Am. J. Med.* 108, 652–659.

Meng, C., Jin, S., Wang, L., Guo, F., and Zou, Q. (2019). AOPs-SVM: a sequence-based classifier of antioxidant proteins using a support vector machine. *Front. Bioeng. Biotechnol.* 7:224. doi: 10.3389/fbioe.2019.00224

Mirny, L. A., and Shakhnovich, E. I. (1999). Universally conserved positions in protein folds: reading evolutionary signals about stability, folding kinetics and function. *J. Mol. Biol.* 291, 177–196. doi: 10.1006/jmbi.1999.2911

Nichole, C., Ying, Z., Marian, N., and Fereidoon, S. (2008). Antioxidant activity and water-holding capacity of canola protein hydrolysates. *Food Chem.* 109, 144–148. doi: 10.1016/j.foodchem.2007.12.039

Solis, A. D., and Rackovsky, S. (2000). Optimized representations and maximal information in proteins. *Proteins* 38, 149–164. doi: 10.1002/(sici)1097-0134(20000201)38:2<149::aid-prot4>3.0.co;2-#

Tan, J. X., Li, S. H., Zhang, Z. M., Chen, C. X., Chen, W., Tang, H., et al. (2019). Identification of hormone binding proteins based on machine learning methods. *Math. Biosci. Eng.* 16, 2466–2480. doi: 10.3934/mbe.2019123

Thomas, P. D., and Dill, K. A. (1996). An iterative method for extracting energy-like quantities from protein structures. *Proc. Natl. Acad. Sci. U.S.A.* 93, 11628–11633. doi: 10.1073/pnas.93.21.11628

Wang, J., and Wang, W. (1999). A computational approach to simplifying the protein folding alphabet. *Nat. Struct. Biol.* 6, 1033–1038.

Wang, Y., Shi, F., Cao, L., Dey, N., Wu, Q., Ashour, A. S., et al. (2019). Morphological segmentation analysis and texture-based support vector machines classification on mice liver fibrosis microscopic images. *Curr. Bioinform.* 14, 282–294. doi: 10.2174/1574893614666190304125221

Wei, L., Tang, J., and Zou, Q. (2017). Local-DPP: an improved DNA-binding protein prediction method by exploring local evolutionary information. *Inform. Sci.* 384, 135–144. doi: 10.1016/j.ins.2016.06.026

Yang, S., Lulu, W., Ying, W., Xiaoqian, O., Zhaoyuan, S., Chongchong, L., et al. (2017). Purification and identification of a natural antioxidant protein from fertilized eggs. *Korea. J. Food Sci. Anim. Resourc.* 37, 764–772. doi: 10.5851/kosfa.2017.37.5.764

Yang, W., Zhu, X., Huang, J., Ding, H., and Lin, H. (2019). A brief survey of machine learning methods in protein sub-golgi localization. *Curr. Bioinform.* 13, 234–240. doi: 10.2174/1574893613666181113131415

Yigit, A. A., Panda, A. K., and Cherian, G. (2014). The avian embryo and its antioxidant defence system. *Worlds Poul. Sci. J.* 70, 563–574. doi: 10.1017/s0043933914000610

Zheng, L., Huang, S., Mu, N., Zhang, H., Zhang, J., Chang, Y., et al. (2019). RAACBook: a web server of reduced amino acid alphabet for sequence-dependent inference by using Chou's five-step rule. *Database* 2019:baz131.

Zuo, Y., Li, Y., Chen, Y., Li, G., Yan, Z., and Yang, L. (2017). PseKRAAC: a flexible web server for generating pseudo K-tuple reduced amino acids composition. *Bioinformatics* 33, 122–124. doi: 10.1093/bioinformatics/btw564

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 12 August 2020
doi: 10.3389/fbioe.2020.00892

Prediction of Anticancer Peptides Using a Low-Dimensional Feature Model

Qingwen Li[1]†, Wenyang Zhou[2]†, Donghua Wang[3]*, Sui Wang[4,5]* and Qingyuan Li[6]*

[1] College of Animal Science and Technology, Northeast Agricultural University, Harbin, China, [2] Center for Bioinformatics, School of Life Sciences and Technology, Harbin Institute of Technology, Harbin, China, [3] Department of General Surgery, Heilongjiang Province Land Reclamation Headquarters General Hospital, Harbin, China, [4] Key Laboratory of Soybean Biology in Chinese Ministry of Education, Northeast Agricultural University, Harbin, China, [5] State Key Laboratory of Tree Genetics and Breeding, Northeast Forestry University, Harbin, China, [6] Forestry and Fruit Tree Research Institute, Wuhan Academy of Agricultural Sciences, Wuhan, China

OPEN ACCESS

Edited by:
Zhibin Lv,
University of Electronic Science and Technology of China, China

Reviewed by:
Zhao Da,
Huazhong Agricultural University, China
Leyi Wei,
Shandong University, China

***Correspondence:**
Donghua Wang
wangdonghua7885@163.com
Sui Wang
wangsui.ws@163.com
Qingyuan Li
liqingyuan@webmail.hzau.edu.cn

†These authors have contributed equally to this work

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: 25 May 2020
Accepted: 10 July 2020
Published: 12 August 2020

Citation:
Li Q, Zhou W, Wang D, Wang S and Li Q (2020) Prediction of Anticancer Peptides Using a Low-Dimensional Feature Model. Front. Bioeng. Biotechnol. 8:892. doi: 10.3389/fbioe.2020.00892

Cancer is still a severe health problem globally. The therapy of cancer traditionally involves the use of radiotherapy or anticancer drugs to kill cancer cells, but these methods are quite expensive and have side effects, which will cause great harm to patients. With the find of anticancer peptides (ACPs), significant progress has been achieved in the therapy of tumors. Therefore, it is invaluable to accurately identify anticancer peptides. Although biochemical experiments can solve this work, this method is expensive and time-consuming. To promote the application of anticancer peptides in cancer therapy, machine learning can be used to recognize anticancer peptides by extracting the feature vectors of anticancer peptides. Nevertheless, poor performance usually be found in training the machine learning model to utilizing high-dimensional features in practice. In order to solve the above job, this paper put forward a 19-dimensional feature model based on anticancer peptide sequences, which has lower dimensionality and better performance than some existing methods. In addition, this paper also separated a model with a low number of dimensions and acceptable performance. The few features identified in this study may represent the important features of anticancer peptides.

Keywords: anticancer peptide, feature extraction, feature model, feature selection, machine learning

INTRODUCTION

Cancer is still a severe health problem globally, and lots of people have died from cancer (Liao et al., 2018; Cheng et al., 2019a; Zeng W. et al., 2019; Zhang Y. et al., 2019; Zhou et al., 2019; Yang et al., 2020). Traditional cancer treatments kill not only cancer cells but also normal cells, and the medical costs are very high (Feng, 2019; Lin et al., 2019; Li Y.H. et al., 2020; Zhang et al., 2020). With the find of anticancer peptides, the situation has changed because anticancer peptides can interact with the anionic cellular elements of cancer cells to selectively kill cancer cells without harming the normal cells of the body (Ozkan et al., 2019; Wang Y. et al., 2020; Yin et al., 2020). Although there have been some defects in the development of anticancer peptides, anticancer peptides are safer than man-made drugs (Sun et al., 2016; Liu H. et al., 2018; Liao and Jiang, 2019; Munir et al., 2019; Srivastava et al., 2019; Liu H. et al., 2020; Ru et al., 2020;

Wang J. et al., 2020) and have higher effectiveness, specificity and selectivity. Anticancer peptides provide a new direction for the treatment of cancer, so the therapeutic methods of anticancer peptides have attracted greater attention. Anticancer peptides are generally composed of five to thirty amino acids. Nevertheless, it is still hard to identify anticancer peptides from other (artificially designed or natural) peptides. Using biochemical experiments to identify anticancer peptides is very time-consuming and expensive. In addition, only a few anticancer peptides can be used in the clinic. Thus, it is essential to apply machine learning to forecast anticancer peptides.

In past few years, some bioinformatics methods have been introduced to predict anticancer peptides. By extracting the amino acid composition and binary features of anticancer peptides as feature vectors, Tyagi et al. (2013) applied support vector machine to verify the performance, and the accuracy reached 91.44%. Hajisharifi et al. (2014) applied support vector machine to predict anticancer peptides on the basis of the local alignment kernel and pseudo-amino acid composition, and the highest accuracy was 89.7%. Chen W. et al. (2016) developed a classifier for predicting anticancer peptides by optimizing the composition of g-GAP dipeptides, and 94.77% accuracy was obtained by using 126D features. Xu et al. (2018b) used 400D features or 400D-g gap features to predict anticancer peptides, and the accuracy of support vector machine reached 91.86%. The above methods obtained sound prediction results, but these methods did not mention the dimensional advantages of the model. In reality, training the machine learning model utilizing high-dimensional features usually behaves poorly, This phenomenon is called Curse of Dimensionality (Wilcox, 1961; Xu et al., 2017; Xu Y. et al., 2018; Zou et al., 2017; Wang et al., 2019).

In this paper, through using a variety of polypeptide feature extraction methods, the obtained feature vectors were selected many times, which gained a low-dimensional model. Using multiple classifiers for verification, the performance accuracy was 92.73%, while the number of dimensions of the model was only 19. In this paper, the most important 7 dimensional features were further separated and verified, and good results were obtained. The feature model obtained in this paper can not only accurately and rapidly classify anticancer peptides, but also effectively avoid Curse of Dimensionality. The above results may suggest that these low-dimensional features are important features for distinguishing anticancer peptides.

MATERIALS AND METHODS

The process of this research is shown in **Figure 1**. Every detailed step will be presented in the following sections.

Benchmark Dataset

In this paper, we used the benchmark dataset constructed by Hajisharifi et al., which contained 206 non-anticancer peptides and 138 anticancer peptides. The anticancer peptides in this data set were extracted from APD2, and 206 non-anticancer peptides established by Wang et al. were extracted from UniProt. To avoid the deviation of the classifier, peptides with more than 90% similarity were deleted from the data set through CD-HIT. Chen et al. and Xu et al. have applied the identical benchmark data set as well.

Feature Extraction Strategies

The peptide sequences can not be immediately identified by machine learning algorithms. Therefore, it is requisite to translate the strings stood for peptide sequences into numerical features (Liu et al., 2006, 2019b; Liu S. et al., 2018; Jia et al., 2018; Wang et al., 2018; Chen C. et al., 2019; Hong J. et al., 2019). The feature extraction methods are very crucial in building computational predictors (Cheng et al., 2018, 2019b; Xiong et al., 2018; Zhang et al., 2018b, 2019a; Sun et al., 2019; Tang et al., 2019).

In this paper, we applied five sorts of feature extraction strategies including amino acid composition (AAC), conjoint triad (CT), pseudo-amino acid composition (PAAC), grouped amino acid composition (GAAC) and C/T/D. Each strategy may also include several feature extraction methods. This paper implemented these strategies through iFeature (Chen et al., 2018).

Conjoint Triad

Shen et al. (2007) put forward the conjoint triad model (CT). In consideration of the properties of one amino acid and its nearby amino acids and regards any three sequential amino acids as a unit, the model classifies amino acids into seven sorts. Triad in the same class are considered similar. As an example, triads which are composed by three amino acids belonging to the same sort, such as GLM and VFT, could be treated equally, since they may play the same role. A peptide sequence is represented by a binary space (V,F). V is the vector space of sequence features. Each feature (v_i) represents a unit. F is the frequency vector corresponding to V, and each feature (f_i) is the frequency of v_i in a peptide sequence.

C/T/D

Dubchak et al. (1995) put forward the C/T/D model. This model considers 3 properties of amino acids, their solubility, secondary structure and relative hydrophobicity. Amino acids are classified into three classes on the basis of the relative hydrophobicity, three or four classes on the basis of the secondary structure, and two classes on the basis of solubility. Each class is presented by the three kinds of descriptors: C/T/D (Tan et al., 2019).

Amino Acid Composition

The peptide is composed of 20 sorts of amino acids (Liu et al., 2019a). The frequency of every amino acid type in a peptide sequence was computed to present the peptide sequences. Therefore, each peptide sequence can be represented as a 20-dimensional feature model. This model is called amino acid composition model (AAC). The features can be defined as:

$$f(a) = {}^{N_a}\!/_{N}, \quad a \in (A, C, \ldots, W, Y)$$

where N_a is the quantity of amino acid type a. while N is the length of a peptide sequence.

In this paper, we also used the k-spaced amino acid pair composition model (CKSAAP), which computes the frequency of amino acid pairs separated by an arbitrary number (k) of

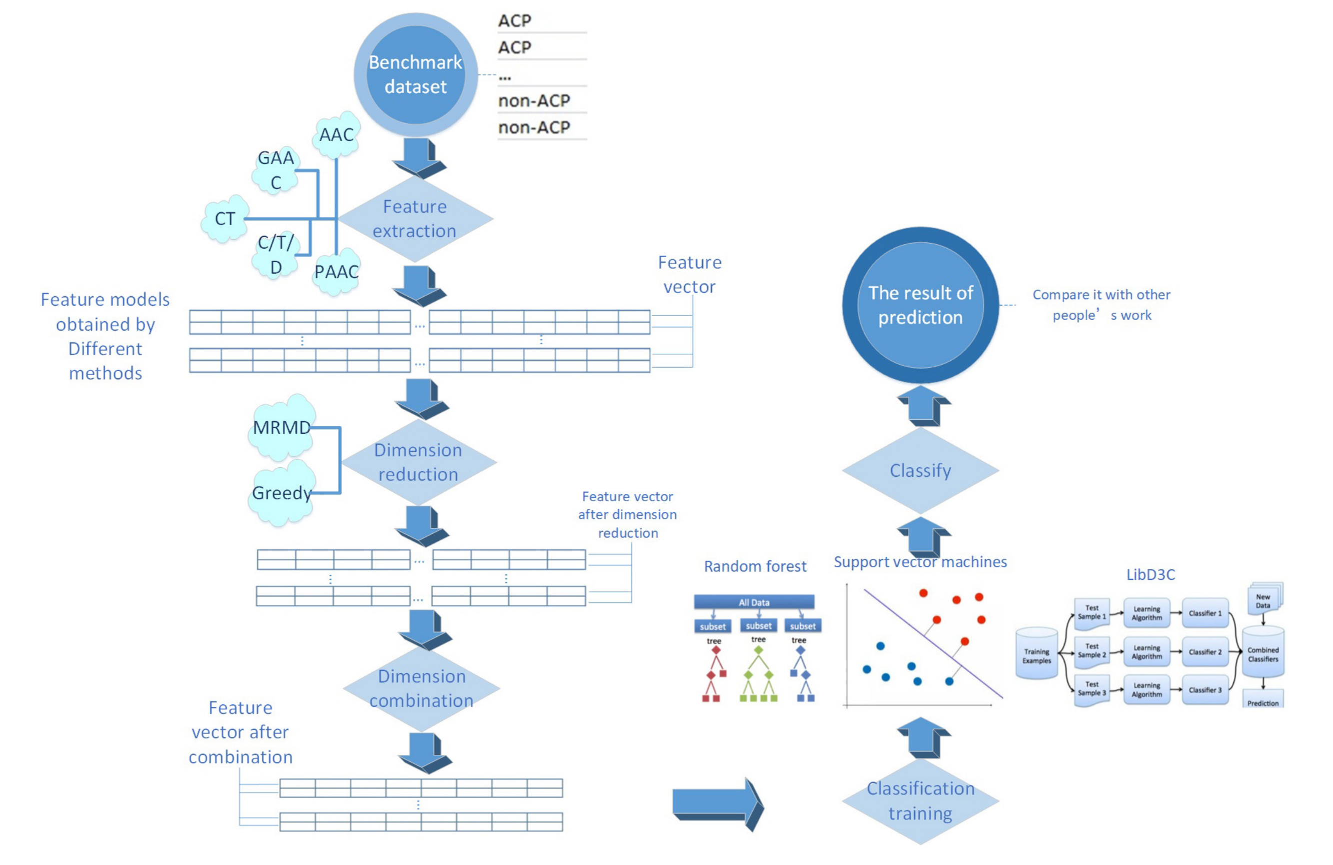


FIGURE 1 | The main flow chart of the research process in this paper.

amino acid residues. A example of this encoding scheme ($k = 0$) is provided as follow:

$$\text{a peptide sequence : CRACRKDSMVN}$$

The features ($k = 0$) can be defined as:

$$\left(N_{AA} = {}^{0}/_{(N-1)}, N_{AC} = {}^{1}/_{(N-1)}, \ldots, N_{CQ} = {}^{0}/_{(N-1)}, N_{CR} = {}^{2}/_{(N-1)}, \ldots, N_{YY} = {}^{0}/_{(N-1)}\right)_{400}$$

At the same time, this paper used the tripeptide composition model (TPC), which computes the frequency of three consecutive amino acids in a peptide sequence and provides 8000 dimensional features. The features can be defined as:

$$f\left(a, b, c\right) = {}^{N_{abc}}/_{(N-2)}, \quad a, b, c \in (A, C, \ldots, W, Y)$$

where N_{abc} is the quantity of amino acid type a, b, and c. while N is the length of a peptide sequence.

At the same time, this paper used the dipeptide composition model (DPC), which computes the frequency of two consecutive amino acids in a peptide sequence and provides 400D features. The features can be defined as:

$$f\left(a, b\right) = {}^{N_{ab}}/_{(N-1)}, \quad a, b \in (A, C, \ldots, W, Y)$$

where N_{ab} is the quantity of amino acid type a and b. while N is the length of a peptide sequence.

Pseudo-Amino Acid Composition

Chou (2001) put forward a pseudo-amino acid composition model (PAAC). In this model, It takes into account not only the frequency of each amino acid type in a peptide sequence but also the position information of the amino acids. Therefore, the feature of the pseudo-amino acid composition is stated as below:

$PAAC = (a_1,a_2,\ldots,a_{19},a_{20},a_{20+1},\ a_{20+2},\ldots,a_{20+n})$

The front portion $a_1,\ldots,\ a_{19},a_{20}$ stand for the frequency of each amino acid type in a peptide sequence, and the latter portion $a_{20+1},\ldots,a_{20+n}$ represent the location info of the amino acids in a peptide sequence.

This paper also used a method similar to PAAC. The amphiphilic pseudo-amino acid composition model (APAAC) was put forward by Chou et al. The model takes the hydrophilic and hydrophobic properties of amino acids into account.

Grouped Amino Acid Composition

The grouped amino acid composition model (GAAC) divides 20 amino acid types into 5 classes on the basis of the physical and chemical properties and then computes the frequency of each amino acid group in a peptide sequence. The features can be defined as:

$$f(c) = {}^{N_c}/_{N}, \quad c \in (c_1, c_2, c_3, c_4, c_5)$$

where N_c is the quantity of amino acid in class c. while N is the length of a peptide sequence.

In this paper, a model similar to the grouped amino acid model, k-spaced amino acid group pair (CKSAAGP), was used to compute the frequency of amino acid group pairs separated by an arbitrary number (k) of amino acid residues.

This paper also used the grouped dipeptide composition model (GDPC), which can be regarded as a combination of GAAC and DPC.

In addition, this paper used the grouped tripeptide composition model (GTPC), which can be regarded as a combination of GAAC and TPC.

Feature Selection

Feature selection is the procedure of picking out a subset from the relevant features applied in machine learning model building (Zou et al., 2016; Qiao et al., 2018; Cheng, 2019; Yang et al., 2019; Zhang M. et al., 2019; Li F. et al., 2020). The dimension of features will be decreased after feature selection, thus this procedure is named dimension reduction as well. MRMD2.0 was mainly used in this paper to reduce the feature dimensions. Each feature was given a numerical value by MRMD2.0 (the larger the number, the feature's recognition ability will be more obvious). MRMD2.0 sorted the features in order on the basis of the ranking value. Next, the first feature with the highest value was examined for its performance. The second feature was added to examine the capability of the new feature subset. This procedure continued till examining total features. Eventually, some parameters in disparate dimensions were acquired, including F-score, accuracy, etc.

Classifier

Support Vector Machine

A support vector machine (SVM) was used for prediction in this study. SVM has been widely applied in the proteome prediction (Jiang et al., 2013; Wei et al., 2016, 2018; Ding et al., 2017; Lin et al., 2017; Qu et al., 2017; Wang et al., 2017, 2018; Guo and Xu, 2018; Xu et al., 2018a,b; Zhang et al., 2018a; Chao et al., 2019; Chen Z. et al., 2019; Fang et al., 2019; Hong Z. et al., 2019; Liu and Li, 2019; Yu and Gao, 2019; Zeng et al., 2019b; Dao et al., 2020; Huang et al., 2020), transcriptome (Chen X. et al., 2016; Tang et al., 2017) and genome (Zeng et al., 2017; Song et al., 2018; Deng et al., 2019b; Hong Z. et al., 2019). Therefore, support vector machine is a pretty useful classifier. libSVM was adopted in this paper to optimize the prediction results of SVM utilizing grid method to correct parameters g and c.

Random Forest

Random forest (rf) has been extensively applied as a classifier in chemoinformatics (Zeng et al., 2019b, 2020a,b; Song et al., 2020) and bioinformatics (Zhang J. et al., 2016; Guo and Xu, 2018; Deng et al., 2019a; Liu et al., 2019a; Lv H. et al., 2019; Lv Z. et al., 2019; Lv et al., 2020; Ru et al., 2019; Wei et al., 2019; Xu et al., 2019; Tang et al., 2020; Yu et al., 2020). Rf was applied in this paper.

LibD3C

At the same time, this paper used the LibD3C classifier (Lin et al., 2014) for prediction to examine the performance of the model. The classifier adopts the strategy of selective integration, based on the hybrid integrated pruning model on the basis of k-means clustering and functional selection cycle framework and sequential search, by training multiple classifiers and selecting a group of accurate and diversified classifiers to solve the problem.

Prediction Result Estimate

It is extremely critical to quantitatively evaluate the effectiveness of the method because the benchmark data set is non-balanced data. This paper used Mathew correlation coefficient (Mcc), specificity (Sp),sensitivity (Sn), total accuracy (Acc) and the F-score value (F-score) phase to evaluate the performance of the model (Li et al., 2015, 2017; Wei et al., 2017; Chu et al., 2019; Ding et al., 2019; Gong et al., 2019; Liang et al., 2019; Shan et al., 2019; Yan et al., 2019; Yu and Gao, 2019; Zeng et al., 2019a, 2020b; Zhang et al., 2019b; Liu X. et al., 2020; Wang H. et al., 2020).

$$\mathrm{Mcc} = (\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}) /$$

$$\sqrt{(\mathrm{TP} + \mathrm{FP}) \times (\mathrm{TP} + \mathrm{FN}) \times (\mathrm{TN} + \mathrm{FP}) \times (\mathrm{TN} + \mathrm{FN})}$$

$$\mathrm{Sn} = \mathrm{TP}/\,(\mathrm{TP} + \mathrm{FN})$$

$$\mathrm{Sp} = \mathrm{TN}/\,(\mathrm{TN} + \mathrm{FP})$$

$$\mathrm{Acc} = (\mathrm{TP} + \mathrm{TN})\,/\,(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})$$

$$\mathrm{F} - \mathrm{score} = 2 \times \mathrm{P} \times \mathrm{R}/\,(\mathrm{P} + \mathrm{R})$$

where TP stands for the quantity of anticancer peptides correctly predicted, FP stands for the quantity of non-anticancer peptides predicted as anticancer peptides, TN stands for the correctly predicted quantity of non-anticancer peptides, and FN stands for the quantity of anticancer peptides predicted as non-anticancer peptides. P represents the accuracy, indicating the proportion of the total number of predicted positive cases; R is the recall rate, indicating the number of correct cases identified and accounting for the total number of cases in this category.

RESULTS AND DISCUSSION

In this paper, a total of 12 feature extraction methods were used. Because the number of dimensions of the amino acid composition model was only 20, it is of little significance to reduce the dimensionality of the amino acid composition model alone, and the k-spaced amino acid pair composition model is an extension of this method. The principles of the two models were similar, and so the two models were merged and expressed uniformly by AAC. Similarly, the grouped amino acid composition model and the k-spaced amino acid group pair model were merged and expressed uniformly by GAAC. To compare the advantages and disadvantages of different feature extraction methods for anticancer peptide sequences, each model obtained by each method was examined by 10-fold cross-validation utilizing the random forest classifier, and then 10-fold cross-validation was carried out for each method after dimensional reduction through MRMD2.0. **Figure 2A** lists the F-score of each feature extraction method before and after feature

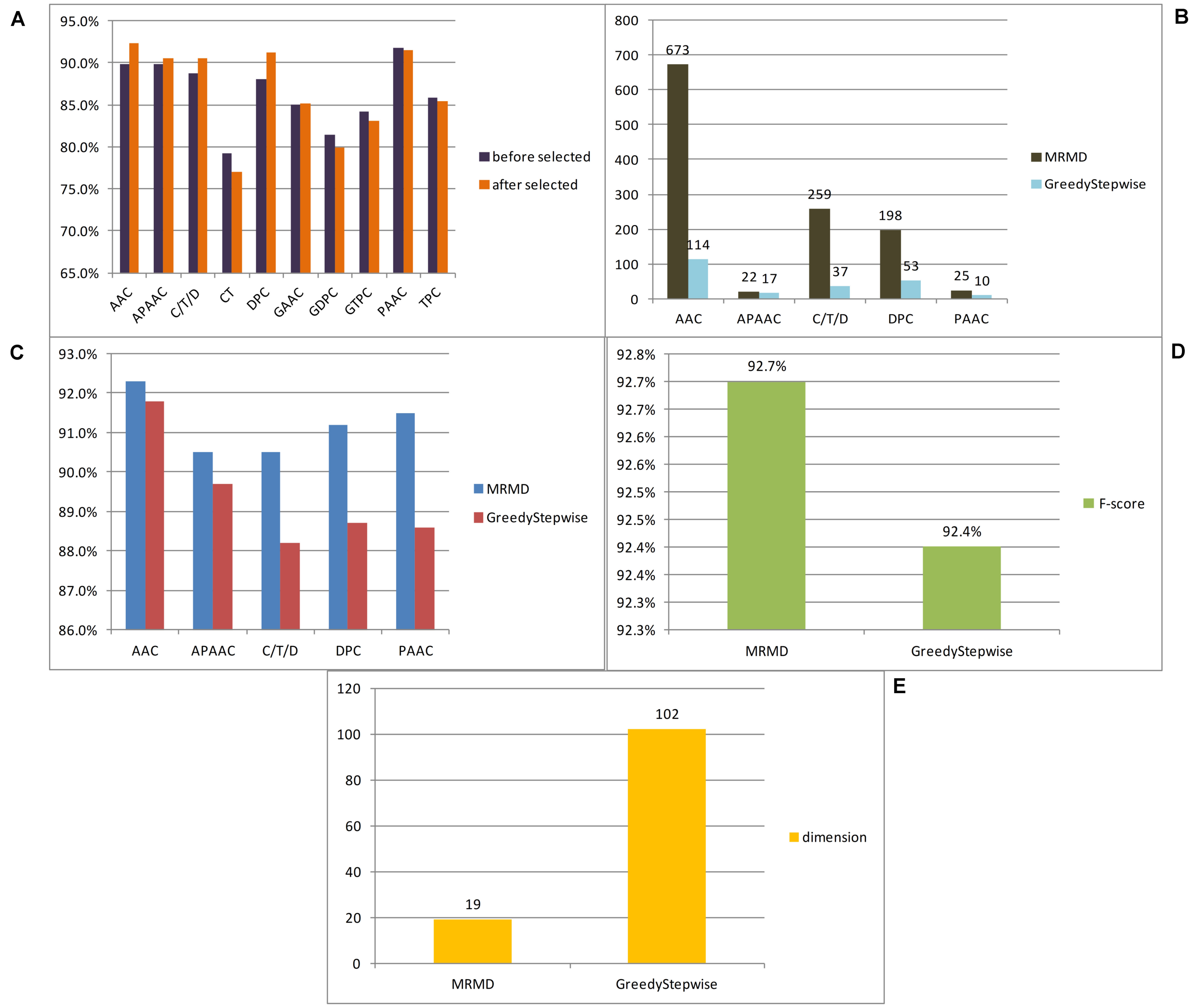


FIGURE 2 | The results of different experiments. **(A)** According to the results, this paper thought that the CT, GAAC, GDPC, GTPC, and TPC are not ideal. **(B)** According to the results, this paper thought that the greedy algorithm was more efficient than MRMD2.0. **(C)** According to the results, this paper thought that the greedy algorithm is worse than MRMD2.0 in the performance index of the selected model. **(D)** After several dimension reductions, the results showed that the MRMD2.0 was better than the greedy algorithm index of the selected model. **(E)** After several dimension reductions, the results showed that the dimension of model of the greedy algorithm is about five times that of the MRMD2.0. The results showed that as for the dimensions of the selected model, the greedy algorithm was more efficient than MRMD2.0. However, the greedy algorithm cannot further reduce the dimensions of the selected feature model, but MRMD2.0 can still further reduce it.

TABLE 1 | Comparing the performance of different methods.

Methods	Sn	Sp	Acc	MCC	F-score	Dimension
iACP	88.40%	99.02%	94.77%	89.30%		126
Hajisharifi et al.	85.18%	92.68%	89.70%	78.40%		
SAP	86.23%	95.63%	91.86%	83.01%	89.47%	400
Our method(RF)	86.20%	97.10%	92.73%	84.90%	92.70%	19
Our method(LibD3C)	85.50%	96.60%	92.15%	83.70%	92.10%	19
Our method(SVM)	87.70%	96.10%	92.73%	84.80%	92.70%	19

selection. In this paper, according to the verification results, it is believed that the effects of the CT, GAAC, GDC, GTC, and TC methods were not ideal, so the above model was not considered in the follow-up study. To compare the advantages and disadvantages of different feature selection methods, the greedy algorithm and MRMD2.0 were used to select each feature model. **Figure 2B** lists the dimensions of each feature model after two kinds of software selection, and **Figure 2C** lists the F-score of each feature model after two kinds of software selection. For the feature selection method of anticancer peptide, after synthesizing the situation of all types of model selection, MRMD2.0 was better than the greedy algorithm in terms of the capability index of the selected model; As for the dimensions of the selected model, the greedy algorithm was more efficient than MRMD2.0. However, the greedy algorithm cannot further reduce the dimensions of the selected feature model, but MRMD2.0 can still further reduce it.

The feature subset of each method was merged and reduced to get a 102D feature model after selected by the greedy algorithm. The F-score value was 0.924 after random forest 10-fold cross-validation. At this time, it was impossible to use the greedy algorithm to further reduce the dimensions of the model.

After merging the selected feature model by MRMD2.0, the model dimension number was 1177. This paper continued to use MRMD2.0 to reduce the dimension of the model to get a 767-dimensional feature model which was still too high. After continuing to reduce the dimensionality of the model again to obtain 633 dimensional features, the result was still not ideal. In this paper, the dimensionality reduction was carried out 6 times. For each dimensionality reduction, a line chart of F-score was drawn changing with the dimension according to the obtained indicators. The feature points were separated with large changes in the line to form a new model for verification, and the results were not ideal. After 8 times of dimensionality reduction, a 19-dimensional feature model was obtained. At this time, it was no longer possible to use MRMD2.0 for dimensionality reduction. **Figures 2D,E** list the feature model F-score and dimensions separated by the two methods, respectively. By comparison, MRMD2.0 was found to be better than the greedy algorithm.

The 19-dimensional model was tested by random forest, support vector machine (parameters c and g are 8192.0 and 0.00048828125, respectively) and LibD3C, respectively. **Table 1** listed the prediction results of three types of classifiers. The results indicated that the performance of the 19-dimensional model separated in this paper is stable. **Table 1** also lists the prediction results of others based on the same data set. Compared with Hajisharifi et al.'s and Xu et al.'s models, the model in this paper performs better in all prediction indicators. Although it is slightly inferior to Chen et al. in the prediction results, the number of dimensions of their model was 126, while the number of dimensions of this paper is 19, which is obviously lower than that in the previous study. By evaluating the performance

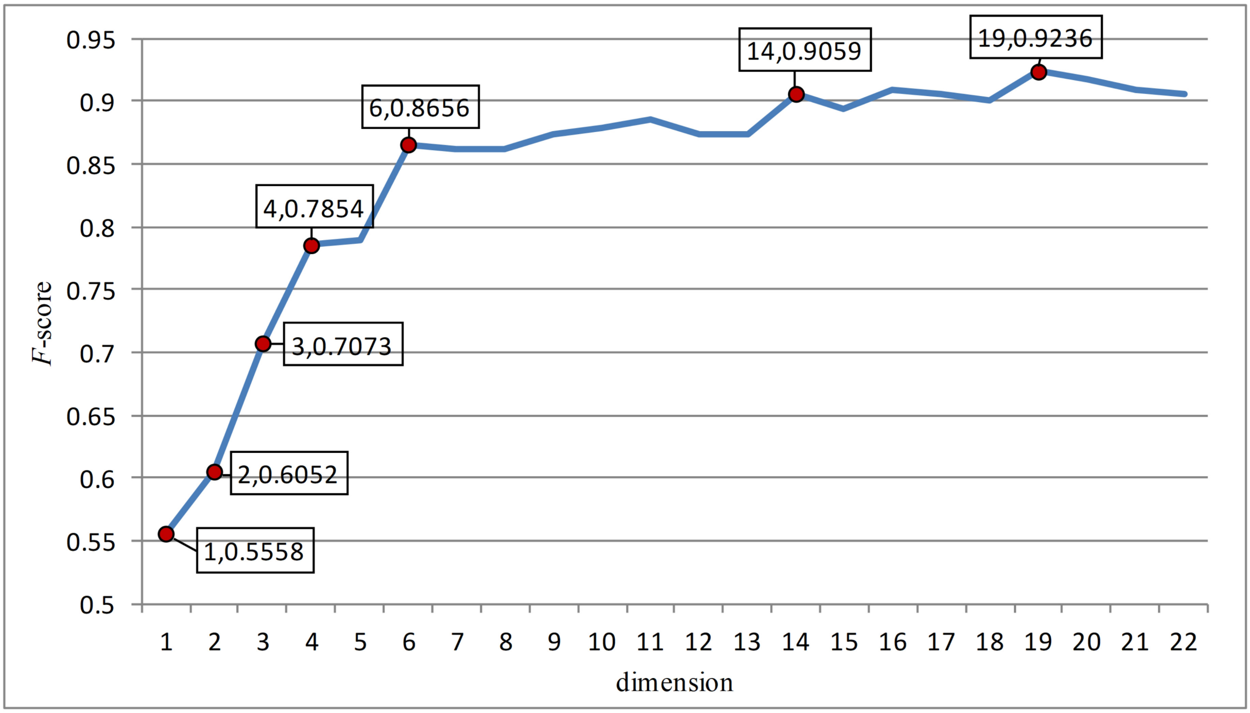


FIGURE 3 | The figure was the change of F-score with dimension according to the last dimension reduction. The red dots in the figure were the feature points with great changes in this paper. And these points were separated to form a new feature model and verified. After verification, these seven red dots are the most important seven features.

of the model and comparing it with the previous work, this paper believed that the 19-dimensional model proposed in this paper can be used to predict the anticancer peptide conveniently, quickly and accurately.

In this paper, the feature points with large slopes in the last reduced-dimension line chart (**Figure 3**) were separated to form a 7-dimensional model, which was verified by support vector machine with an accuracy of 90.41%. This possibly imply that these seven-dimensional features are important features to distinguish anticancer peptides. These 7-dimensional features are GL.gap4, hydrophobicity_PRAM900101.Tr2332, polarizability.2.residue0, Pc1.C, Xc1.K, Pc2.Hydrophobicity.8, and secondarystruct.1.residue0. These features may suggest that for anticancer peptides, the composition and content of glycine, leucine, cysteine and lysine as well as their secondary structure, polarization and hydrophobicity are important indicators different from other non-anticancer peptides.

CONCLUSION

In this paper, a low-dimensional feature model with better performance was obtained through feature extraction and continuous feature selection over many iterations. The features were further isolated, and a few features that might distinguish anticancer peptides were identified. It is hoped that the results of this paper can be used in the artificial design and prediction of anticancer peptides.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

QWL and SW conceived and designed the research. QWL and WZ performed the machine learning experiments. QWL, DW, and QYL analyzed the data. QWL and WZ wrote the manuscript. QYL and SW coordinated the study and revised the manuscript. All authors read and approved the final manuscript.

ACKNOWLEDGMENTS

This manuscript used iFeature online tool to extract features, used random forest classifier through Weka platform, and used MRMD2.0 to reduce dimensions. Yuwei Jiang and Dongyuan Yu contributed to the language editing of this article. Yuwei Jiang and Dongyuan Yu are from Tianjin Normal University and Northeast Agricultural University, respectively.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fbioe.2020.00892/full#supplementary-material

REFERENCES

Chao, L., Wei, L., and Zou, Q. (2019). SecProMTB: a SVM-based classifier for secretory proteins of mycobacterium tuberculosis with imbalanced data set. *Proteomics* 19:e1900007.

Chen, C., Zhang, Q., Ma, Q., and Yu, B. (2019). LightGBM-PPI: predicting protein-protein interactions through LightGBM with multi-information fusion. *Chemometr. Intellig. Lab. Syst.* 191, 54–64. doi: 10.1016/j.chemolab.2019.06.003

Chen, Z., Zhao, P., Li, F., Marquez-Lago, T., Leier, A., Revote, J., et al. (2019). iLearn: an integrated platform and meta-learner for feature engineering, machine learning analysis and modeling of DNA, RNA and protein sequence data. *Briefings Bioinform.* 21, 1047–1057. doi: 10.1093/bib/bbz041

Chen, W., Ding, H., Feng, P., Lin, H., and Chou, K.-C. (2016). iACP: a sequence-based tool for identifying anticancer peptides. *Oncotarget* 7:7815. doi: 10.18632/oncotarget.7815

Chen, X., Pérez-Jiménez, M. J., Valencia-Cabrera, L., Wang, B., and Zeng, X. (2016). Computing with viruses. *Theoret. Computer Sci.* 623, 146–159.

Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T. T., Wang, Y., et al. (2018). iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinform. J.* 34, 2499–2502.

Cheng, L. (2019). Computational and biological methods for gene therapy. *Curr. Gene Ther.* 19, 210–210.

Cheng, L., Jiang, Y., Ju, H., Sun, J., Peng, J., Zhou, M., et al. (2018). InfAcrOnt: calculating cross-ontology term similarities using information flow by a random walk. *BMC Genomics* 19(Suppl. 1):919. doi: 10.1186/s12864-017-4338-6

Cheng, L., Yang, H., Zhao, H., Pei, X., Shi, H., Sun, J., et al. (2019a). MetSigDis: a manually curated resource for the metabolic signatures of diseases. *Brief Bioinform.* 20, 203–209. doi: 10.1093/bib/bbx103

Cheng, L., Zhao, H., Wang, P., Zhou, W., Luo, M., Li, T., et al. (2019b). Computational methods for identifying similar diseases. *Mol. Ther. Nucleic Acids* 18, 590–604.

Chou, K.-C. (2001). Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins* 43, 246–255. doi: 10.1002/prot.1035

Chu, Y., Kaushik, A. C., Wang, X., Wang, W., Zhang, Y., Shan, X., et al. (2019). DTI-CDF: a cascade deep forest model towards the prediction of drug-target interactions based on hybrid features. *Brief Bioinform.* 2019:bbz152. doi: 10.1093/bib/bbz152

Dao, F. Y., Lv, H., Zulfiqar, H., Yang, H., Su, W., Gao, H., et al. (2020). A computational platform to identify origins of replication sites in eukaryotes. *Brief Bioinform*. 2020:bbaa017. doi: 10.1093/bib/bbaa017

Deng, L., Li, W., and Zhang, J. (2019a). "LDAH2V: Exploring meta-paths across multiple networks for lncRNA-disease association prediction," in *Proceedings of the IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Piscataway, NJ.

Deng, L., Wang, J., and Zhang, J. (2019b). Predicting gene ontology function of human micrornas by integrating multiple networks. *Front. Genet.* 10:3. doi: 10.3389/fmicb.2018.0003

Ding, Y., Tang, J., and Guo, F. (2017). Identification of drug-target interactions via multiple information integration. *Inform. Sci.* 418–419, 546–560. doi: 10.1016/j.ins.2017.08.045

Ding, Y., Tang, J., and Guo, F. (2019). Identification of drug-side effect association via multiple information integration with centered kernel alignment. *Neurocomputing* 325, 211–224. doi: 10.1016/j.neucom.2018.10.028

Dubchak, I., Muchnik, I., Holbrook, S. R., and Kim, S. H. (1995). Prediction of protein folding class using global description of amino acid sequence. *Proc. Natl. Acad. Sci. U.S.A.* 92, 8700–8704. doi: 10.1073/pnas.92.19.8700

Fang, T., Zhang, Z., Sun, R., Zhu, L., He, J., Huang, B., et al. (2019). RNAm5CPred: prediction of RNA 5-methylcytosine sites based on three different kinds of nucleotide composition. *Mol. Ther. Nucleic Acids* 18, 739–747. doi: 10.1016/j.omtn.2019.10.008

Feng, Y. M. (2019). Gene therapy on the road. *Curr. Gene Ther.* 19:6. doi: 10.2174/1566523219999190426144513

Gong, Y., Niu, Y., Zhang, W., and Li, X. (2019). A network embedding-based multiple information integration method for the MiRNA-disease association prediction. *BMC Bioinform.* 20:468. doi: 10.1186/s12859-019-3063-3

Guo, M., and Xu, Y. (2018). Single-cell transcriptome analysis using SINCERA pipeline *Transcriptome. Data Analy.* 1751, 209–222.

Hajisharifi, Z., Piryaiee, M., Mohammad Beigi, M., Behbahani, M., and Mohabatkar, H. (2014). Predicting anticancer peptides with Chou's pseudo amino acid composition and investigating their mutagenicity via Ames test. *J. Theor. Biol.* 341, 34–40. doi: 10.1016/j.jtbi.2013.08.037

Hong, J., Luo, Y., Zhang, Y., Ying, J., Xue, W., Xie, T., et al. (2019). Protein functional annotation of simultaneously improved stability, accuracy and false discovery rate achieved by a sequence-based deep learning. *Brief Bioinform.* 21, 1437–1447. doi: 10.1093/bib/bbz081

Hong, Z., Zeng, X., Wei, L., and Liu, X. J. B. (2019). Identifying enhancer-promoter interactions with neural network based on pre-trained DNA vectors and attention mechanism. *Bioinformatics* 36, 1037–1043.

Huang, Q., Chen, Y., Liu, L., Tao, D., and Li, X. (2020). On combining biclustering mining and adaboost for breast tumor classification. *IEEE Trans. Knowl. Data Eng.* 32, 728–738. doi: 10.1109/TKDE.2019.2891622

Jia, C. Z., Zuo, Y., and Zou, Q. (2018). O-GlcNAcPRED-II: an integrated classification algorithm for identifying O-GlcNAcylation sites based on fuzzy undersampling and a K-means PCA oversampling technique. *Bioinformatics* 34, 2029–2036. doi: 10.1093/bioinformatics/bty039

Jiang, Q. H., Wang, G. H., Jin, S. L., Li, Y., and Wang, Y. D. (2013). Predicting human microRNA-disease associations based on support vector machine. *Intern. J. Data Min. Bioinform.* 8, 282–293. doi: 10.1504/ijdmb.2013.056078

Li, B., Tang, J., Yang, Q., Li, S., Cui, X., Li, Y., et al. (2017). NOREVA: normalization and evaluation of MS-based metabolomics data. *Nucleic Acids Res.* 45, W162–W170. doi: 10.1093/nar/gkx449

Li, F., Zhou, Y., Zhang, X., Tang, J., Yang, Q., Zhang, Y., et al. (2020). SSizer: determining the sample sufficiency for comparative biological study. *J. Mol. Biol.* 432:3411. doi: 10.1016/j.jmb.2020.01.027

Li, Y. H., Li, X. X., Hong, J. J., Wang, Y. X., Fu, J. B., Yang, H., et al. (2020). Clinical trials, progression-speed differentiating features and swiftness rule of the innovative targets of first-in-class drugs. *Brief Bioinform.* 21, 649–662. doi: 10.1093/bib/bby130

Li, W., Yu, J., Lian, B., Sun, H., Li, J., Zhang, M., et al. (2015). Identifying prognostic features by bottom-up approach and correlating to drug repositioning. *PLoS One* 10:e0118672. doi: 10.1371/journal.pone.0118672

Liang, C., Changlu, Q., He, Z., Tongze, F., and Xue, Z. (2019). gutMDisorder: a comprehensive database for dysbiosis of the gut microbiota in disorders and interventions. *Nucleic Acids Res.* 48, D554–D560.

Liao, Y.-D., and Jiang, Z.-R. (2019). MoABank: an integrated database for drug mode of action knowledge. *Curr. Bioinform.* 14, 446–449. doi: 10.2174/1574893614666190416151344

Liao, Z. J., Li, D. P., Wang, X. R., Li, L. S., and Zou, Q. (2018). Cancer diagnosis through isomir expression with machine learning method. *Curr. Bioinform.* 13, 57–63. doi: 10.2174/1574893611666160609081155

Lin, C., Chen, W., Qiu, C., Wu, Y., Krishnan, S., and Zou, Q. (2014). LibD3C: ensemble classifiers with a clustering and dynamic selection strategy. *Neurocomputing* 123, 424–435. doi: 10.1016/j.neucom.2013.08.004

Lin, H., Liang, Z. Y., Tang, H., and Chen, W. (2017). Identifying sigma70 promoters with novel pseudo nucleotide composition. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 16, 1316–1321. doi: 10.1109/TCBB.2017.2666141

Lin, M., Li, X., Guo, H., Ji, F., Ye, L., Ma, X., et al. (2019). Identification of bone metastasis-associated genes of gastric cancer by genome-wide transcriptional profiling. *Curr. Bioinform.* 14, 62–69. doi: 10.2174/1574893612666171121154017

Liu, B., Chen, S., Yan, K., and Weng, F. (2019a). iRO-PsekGCC: identify DNA replication origins based on pseudo k-tuple GC composition. *Front. Genet.* 10:842. doi: 10.3389/fmicb.2018.0842

Liu, B., Gao, X., and Zhang, H. (2019b). BioSeq-Analysis2.0: an updated platform for analyzing DNA, RNA, and protein sequences at sequence level and residue level based on machine learning approaches. *Nucleic Acids Res.* 47:e127.

Liu, B., and Li, K. (2019). iPromoter-2L2.0: identifying promoters and their types by combining smoothing cutting window algorithm and sequence-based features. *Mol. Ther.Nucleic Acids* 18, 80–87.

Liu, H., Luo, L. B., Cheng, Z. Z., Sun, J. J., Guan, J. H., Zheng, J., et al. (2018). Group-sparse modeling drug-kinase networks for predicting combinatorial drug sensitivity in cancer cells. *Curr. Bioinform.* 13, 437–443. doi: 10.2174/1574893613666180118104250

Liu, S., Liu, C., and Deng, L. (2018). Machine learning approaches for protein-protein interaction hot spot prediction: progress and comparative assessment. *Molecules* 23:2535.

Liu, H., Zhang, W., Zou, B., Wang, J., Deng, Y., and Deng, L. (2020). DrugCombDB: a comprehensive database of drug combinations toward the discovery of combinatorial therapy. *Nucleic Acids Res.* 48, D871–D881.

Liu, X., Hong, Z., Liu, J., Lin, Y., Rodríguez-Patón, A., Zou, Q., et al. (2020). Computational methods for identifying the critical nodes in biological networks. *Briefings Bioinform.* 21, 486–497.

Liu, W., Meng, X., Xu, Q., Flower, D. R., and Li, T. (2006). Quantitative prediction of mouse class I MHC peptide binding affinity using support vector machine regression (SVR) models. *BMC Bioinform.* 7:182. doi: 10.1186/1471-2105-7-182

Lv, H., Dao, F.-Y., Zhang, D., Guan, Z.-X., Yang, H., Su, W., et al. (2020). iDNA-MS: an integrated computational tool for detecting DNA modification sites in multiple genomes. *iScience* 23:100991. doi: 10.1016/j.isci.2020.100991

Lv, H., Zhang, Z. M., Li, S. H., Tan, J. X., Chen, W., and Lin, H. (2019). Evaluation of different computational methods on 5-methylcytosine sites identification. *Briefings Bioinform.* 21, 982–995. doi: 10.1093/bib/bbz048

Lv, Z., Jin, S., Ding, H., and Zou, Q. (2019). A random forest sub-Golgi protein classifier optimized via dipeptide and amino acid composition features. *Front. Bioeng. Biotechnol.* 7:215. doi: 10.3389/fmicb.2018.00215

Munir, A., Malik, S. I., and Malik, K. A. (2019). Proteome mining for the identification of putative drug targets for human pathogen clostridium tetani. *Curr. Bioinform.* 14, 532–540. doi: 10.2174/1574893613666181114095736

Ozkan, A., Isgor, S. B., Sengul, G., and Isgor, Y. G. (2019). Benchmarking classification models for cell viability on novel cancer image datasets. *Curr. Bioinform.* 14, 108–114. doi: 10.2174/1574893614666181120093740

Qiao, Y., Xiong, Y., Gao, H., Zhu, X., and Chen, P. (2018). Protein-protein interface hot spots prediction based on a hybrid feature selection strategy. *BMC Bioinform.* 19:14. doi: 10.1186/s12859-018-2009-5

Qu, K., Han, K., Wu, S., Wang, G., and Wei, L. (2017). Identification of DNA-binding proteins using mixed feature representation methods. *Molecules* 22:1602. doi: 10.3390/molecules22101602

Ru, X., Wang, L., Li, L., Ding, H., Ye, X., and Zou, Q. (2020). Exploration of the correlation between GPCRs and drugs based on a learning to rank algorithm. *Comput. Biol. Med.* 119:103660.

Ru, X. Q., Li, L. H., and Zou, Q. (2019). Incorporating Distance-based top-n-gram and random forest to identify electron transport proteins. *J. Proteome Res.* 18, 2931–2939. doi: 10.1021/acs.jproteome.9b00250

Shan, X., Wang, X., Li, C. D., Chu, Y., Zhang, Y., Xiong, Y., et al. (2019). Prediction of CYP450 enzyme-substrate selectivity based on the network-based label space division method. *J. Chem. Inf. Model.* 59, 4577–4586. doi: 10.1021/acs.jcim.9b00749

Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., et al. (2007). Predicting protein-protein interactions based only on sequences information. *Proc. Natl. Acad. Sci. U.S.A.* 104, 4337–4341. doi: 10.1073/pnas.0607879104

Song, B., Li, K., Orellana-Martín, D., Valencia-Cabrera, L., and Pérez-Jiménez, M. J. (2020). Cell-like P systems with evolutional symport/antiport rules and membrane creation. *Inform. Comput.* 2020:104542.

Song, T., Rodríguez-Patón, A., Zheng, P., and Zeng, X. (2018). Spiking neural P systems with colored spikes. *IEEE Trans. Cogn. Dev. Syst.* 10, 1106–1115.

Srivastava, N., Mishra, B. N., and Srivastava, P. (2019). In-silico identification of drug lead molecule against pesticide exposed-neurodevelopmental disorders through network-based computational model approach. *Curr. Bioinform.* 14, 460–467. doi: 10.2174/1574893613666181112130346

Sun, Y., Zhang, W., Chen, Y., Ma, Q., Wei, J., and Liu, Q. (2016). Identifying anti-cancer drug response related genes using an integrative analysis of transcriptomic and genomic variations with cell line-based drug perturbations. *Oncotarget* 7:9404.

Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. (2019). Rotate: knowledge graph embedding by relational rotation in complex space. *arXiv* [Preprint]. arXiv:1902.10197v1

Tan, J. X., Li, S. H., Zhang, Z. M., Chen, C. X., Chen, W., Tang, H., et al. (2019). Identification of hormone binding proteins based on machine learning methods. *Math. Biosci. Eng.* 16, 2466–2480. doi: 10.3934/mbe.2019123

Tang, J., Fu, J., Wang, Y., Li, B., Li, Y., Yang, Q., et al. (2020). ANPELA: analysis and performance assessment of the label-free quantification workflow for metaproteomic studies. *Brief Bioinform.* 21, 621–636. doi: 10.1093/bib/bby127

Tang, J., Fu, J., Wang, Y., Luo, Y., Yang, Q., Li, B., et al. (2019). Simultaneous improvement in the precision, accuracy, and robustness of label-free proteome quantification by optimizing data manipulation chains. *Mol. Cell Proteom.* 18, 1683–1699. doi: 10.1074/mcp.RA118.001169

Tang, Y., Liu, D., Wang, Z., Wen, T., and Deng, L. (2017). A boosting approach for prediction of protein-RNA binding residues. *BMC Bioinform.* 18:465. doi: 10.1186/s12859-018-2009-465

Tyagi, A., Kapoor, P., Kumar, R., Chaudhary, K., Gautam, A., and Raghava, G. P. (2013). In silico models for designing and discovering novel anticancer peptides. *Sci. Rep.* 3:2984. doi: 10.1038/srep02984

Wang, H., Ding, Y., Tang, J., and Guo, F. (2020). Identification of membrane protein types via multivariate information fusion with Hilbert-Schmidt Independence criterion. *Neurocomputing* 383, 257–269. doi: 10.1016/j.neucom.2019.11.103

Wang, J., Wang, H., Wang, X., and Chang, H. (2020). Predicting drug-target interactions via FM-DNN learning. *Curr. Bioinform.* 15, 68–76. doi: 10.2174/1574893614666190227160538

Wang, Y., Zhang, S., Li, F., Zhou, Y., Zhang, Y., Wang, Z., et al. (2020). Therapeutic target database 2020: enriched resource for facilitating research and early development of targeted therapeutics. *Nucleic Acids Res.* 48, D1031–D1041. doi: 10.1093/nar/gkz981

Wang, X., Yu, B., Ma, A., Chen, C., Liu, B., and Ma, Q. (2018). Protein-protein interaction sites prediction by ensemble random forests with synthetic minority oversampling technique. *Bioinformatics* 35, 2395–2402.

Wang, Y., Ding, Y., Guo, F., Wei, L., and Tang, J. (2017). Improved detection of DNA-binding proteins via compression technology on PSSM information. *PLoS One* 12:185587. doi: 10.1371/journal.pone.0185587

Wang, Y., Liu, K., Ma, Q., Tan, Y., Du, W., Lv, Y., et al. (2019). Pancreatic cancer biomarker detection by two support vector strategies for recursive feature elimination. *Biomark. Med.* 13, 105–121. doi: 10.2217/bmm-2018-0273

Wei, L., Wan, S., Guo, J., and Wong, K. K. (2017). A novel hierarchical selective ensemble classifier with bioinformatics application. *Artif. Intellig. Med.* 83, 82–90.

Wei, L., Zhou, C., Chen, H., Song, J., and Su, R. (2018). ACPred-FL: a sequence-based predictor based on effective feature representation to improve the prediction of anti-cancer peptides. *Bioinformatics* 34, 4007–4016.

Wei, L., Zhou, C., Su, R., and Zou, Q. (2019). PEPred-Suite: improved and robust prediction of therapeutic peptides using adaptive feature representation learning. *Bioinformatics* 35, 4272–4280. doi: 10.1093/bioinformatics/btz246

Wei, L., Zou, Q., Liao, M., Lu, H., and Zhao, Y. (2016). A novel machine learning method for cytokine-receptor interaction prediction. *Combinat. Chem. High Throughput Screen.* 19, 144–152.

Wilcox, R. (1961). Adaptive control processes—A guided tour, by richard bellman, princeton university press, princeton, New Jersey, 1961, 255 pp., $6.50. *Naval Res. Logist. Q.* 8:314. doi: 10.1002/nav.3800080314

Xiong, Y., Wang, Q., Yang, J., Zhu, X., and Wei, D. Q. (2018). PredT4SE-Stack: prediction of bacterial Type IV secreted effectors from protein sequences using a stacked ensemble method. *Front. Microbiol.* 9:2571. doi: 10.3389/fmicb.2018.02571

Xu, L., Liang, G., Liao, C., Chen, G.-D., and Chang, C.-C. (2018a). An efficient classifier for alzheimer's disease genes identification. *Molecules* 23:3140.

Xu, L., Liang, G., Wang, L., and Liao, C. (2018b). A novel hybrid sequence-based model for identifying anticancer peptides. *Genes* 9:158. doi: 10.3390/genes9030158

Xu, Y., Zhao, W., Olson, S. D., Prabhakara, K. S., and Zhou, X. (2018). Alternative splicing links histone modifications to stem cell fate decision. *Genome Biol.* 19, 1–21.

Xu, L., Liang, G., Liao, C., Chen, G.-D., and Chang, C.-C. (2019). k-Skip-n-Gram-RF: a random forest based method for Alzheimer's disease protein identification. *Front. Genet.* 10:33. doi: 10.3389/fgene.2019.00033

Xu, Y., Wang, Y., Luo, J., Zhao, W., and Zhou, X. (2017). Deep learning of the splicing (epi)genetic code reveals a novel candidate mechanism linking histone modifications to ESC fate decision. *Nucleic Acids Res.* 45, 12100–12112. doi: 10.1093/nar/gkx870

Yan, K., Fang, X., Xu, Y., and Liu, B. (2019). Protein fold recognition based on multi-view modeling. *Bioinformatics* 35, 2982–2990.

Yang, Q., Li, B., Tang, J., Cui, X., Wang, Y., Li, X., et al. (2019). Consistent gene signature of schizophrenia identified by a novel feature selection strategy from comprehensive sets of transcriptomic data. *Brief Bioinform.* 21, 1058–1068. doi: 10.1093/bib/bbz049

Yang, Q., Wang, Y., Zhang, Y., Li, F., Xia, W., Zhou, Y., et al. (2020). NOREVA: enhanced normalization and evaluation of time-course and multi-class metabolomic data. *Nucleic Acids Res.* 48, W436–W448. doi: 10.1093/nar/gkaa258

Yin, J., Sun, W., Li, F., Hong, J., Li, X., Zhou, Y., et al. (2020). VARIDT 1.0: variability of drug transporter database. *Nucleic Acids Res* 48, D1042–D1050. doi: 10.1093/nar/gkz779

Yu, L., and Gao, L. (2019). Human pathway-based disease network. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 16, 1240–1249. doi: 10.1109/TCBB.2017.2774802

Yu, L., Xu, F., and Gao, L. (2020). Predict new therapeutic drugs for hepatocellular carcinoma based on gene mutation and expression. *Front. Bioeng. Biotechnol.* 8:8. doi: 10.3389/fbioe.2020.00008

Zeng, W., Wang, F., Ma, Y., Liang, X. C., and Chen, P. (2019). Dysfunctional mechanism of liver cancer mediated by transcription factor and non-coding RNA. *Curr. Bioinform.* 14, 100–107. doi: 10.2174/1574893614666181119121916

Zeng, X., Wang, W., Deng, G., Bing, J., and Zou, Q. (2019a). Prediction of potential disease-associated MicroRNAs by using neural networks. *Mol. Ther. Nucleic Acids* 16, 566–575.

Zeng, X., Zhu, S., Liu, X., Zhou, Y., Nussinov, R., and Cheng, F. (2019b). deepDR: a network-based deep learning approach to in silico drug repositioning. *Bioinformatics* 35, 5191–5198. doi: 10.1093/bioinformatics/btz418

Zeng, X., Liao, Y., Liu, Y., and Zou, Q. (2017). Prediction and validation of disease genes using hetesim scores. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 14, 687–695. doi: 10.1109/tcbb.2016.2520947

Zeng, X., Zhu, S., Hou, Y., Zhang, P., Li, L., Li, J., et al. (2020a). Network-based prediction of drug-target interactions using an arbitrary-order proximity embedded deep forest. *Bioinformatics* 36, 2805–2812. doi: 10.1093/bioinformatics/btaa010

Zeng, X., Zhu, S., Lu, W., Liu, Z., Huang, J., Zhou, Y., et al. (2020b). Target identification among known drugs by deep learning from heterogeneous networks. *Chem. Sci.* 11, 1775–1797. doi: 10.1039/C9SC04336E

Zhang, J., Ju, Y., Lu, H., Xuan, P., and Zou, Q. (2016). Accurate identification of cancerlectins through hybrid machine learning technology. *Int. J. Genom.* 2016:7604641. doi: 10.1155/2016/7604641

Zhang, M., Li, F., Marquez-Lago, T. T., Leier, A., Fan, C., Kwoh, C. K., et al. (2019). MULTiPly: a novel multi-layer predictor for discovering general and specific types of promoters. *Bioinformatics* 35, 2957–2965. doi: 10.1093/bioinformatics/btz016

Zhang, W., Jing, K., Huang, F., Chen, Y., Li, B., Li, J., et al. (2019a). SFLLN: a sparse feature learning ensemble method with linear neighborhood regularization for predicting drug-drug interactions. *Inform. Sci.* 497, 189–201. doi: 10.1016/j.ins.2019.05.017

Zhang, W., Li, Z., Guo, W., Yang, W., and Huang, F. (2019b). "A fast linear neighborhood similarity-based network link inference method to predict microRNA-disease associations," in *Proceedings of the IEEE/ACM Trans Comput Biol Bioinform*, Piscataway, NJ.

Zhang, Y., Kou, C., Wang, S., and Zhang, Y. (2019). Genome-wide differential-based analysis of the relationship between DNA methylation and gene expression in cancer. *Curr. Bioinform.* 14, 783–792. doi: 10.2174/1574893614666190424160046

Zhang, W., Chen, Y., Li, D., and Yue, X. (2018a). Manifold regularized matrix factorization for drug-drug interaction prediction. *J. Biomed. Inform.* 88, 90–97.

Zhang, W., Yue, X., Tang, G., Wu, W., Huang, F., and Zhang, X. (2018b). SFPEL-LPI: sequence-based feature projection ensemble learning for predicting LncRNA-protein interactions. *PLoS Comput. Biol.* 14:e1006616. doi: 10.1371/journal.pcbi.1006616

Zhang, Z. M., Tan, J. X., Wang, F., Dao, F. Y., Zhang, Z. Y., and Lin, H. (2020). Early diagnosis of hepatocellular carcinoma using machine learning method. *Front. Bioeng. Biotechnol.* 8:254. doi: 10.3389/fbioe.2020.00254

Zhou, L. Y., Qin, Z., Zhu, Y. H., He, Z. Y., and Xu, T. (2019). Current RNA-based therapeutics in clinical trials. *Curr. Gene Ther.* 19, 172–196. doi: 10.2174/1566523219666190719100526

Zou, Q., Chen, L., Huang, T., Zhang, Z., and Xu, Y. (2017). Machine learning and graph analytics in computational biomedicine. *Artif. Intell. Med.* 83:1. doi: 10.1016/j.artmed.2017.09.003

Zou, Q., Wan, S., Ju, Y., Tang, J., and Zeng, X. (2016). Pretata: predicting TATA binding proteins with novel features and dimensionality reduction strategy. *BMC Syst. Biol.* 10:114. doi: 10.1186/s12859-018-2009-114

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 22 October 2020
doi: 10.3389/fbioe.2020.584807

Discrimination of Thermophilic Proteins and Non-thermophilic Proteins Using Feature Dimension Reduction

Zifan Guo[1], *Pingping Wang*[2], *Zhendong Liu*[3]* *and Yuming Zhao*[4]*

[1] School of Aeronautics and Astronautic, Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China, [2] School of Life Science and Technology, Harbin Institute of Technology, Harbin, China, [3] School of Computer Science and Technology, Shandong Jianzhu University, Jinan, China, [4] Information and Computer Engineering College, Northeast Forestry University, Harbin, China

OPEN ACCESS

Edited by:
Xue Xu,
Harvard Medical School,
United States

Reviewed by:
Leyi Wei,
Shandong University, China
Xiangxiang Zeng,
Hunan University, China

****Correspondence:***
Zhendong Liu
liuzd2000@126.com
Yuming Zhao
zym@nefu.edu.cn

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *18 July 2020*
Accepted: *11 September 2020*
Published: *22 October 2020*

Citation:
Guo Z, Wang P, Liu Z and Zhao Y (2020) Discrimination of Thermophilic Proteins and Non-thermophilic Proteins Using Feature Dimension Reduction.
Front. Bioeng. Biotechnol. 8:584807.
doi: 10.3389/fbioe.2020.584807

Thermophilicity is a very important property of proteins, as it sometimes determines denaturation and cell death. Thus, methods for predicting thermophilic proteins and non-thermophilic proteins are of interest and can contribute to the design and engineering of proteins. In this article, we describe the use of feature dimension reduction technology and LIBSVM to identify thermophilic proteins. The highest accuracy obtained by cross-validation was 96.02% with 119 parameters. When using only 16 features, we obtained an accuracy of 93.33%. We discuss the importance of the different characteristics in identification and report a comparison of the performance of support vector machine to that of other methods.

Keywords: support vector machine, thermophilic proteins, feature dimension reduction, amino acid, feature selection

INTRODUCTION

Temperature is a critical condition for life. Proteins are less stable than other macromolecules, and temperature changes can easily lead to protein denaturation, which can lead to cell death (Kumar et al., 2000). Thus, it is important to develop a highly efficient method for predicting protein thermophilicity, which will contribute to the design of stable proteins. The properties of many proteins are related to their thermal stability. Studies have shown that the thermal stability of proteins is influenced by ion number, salt bridge presence, amino acid composition (AAC), dipeptide composition (DPC), and other factors (Sadeghi et al., 2006; Wang H. et al., 2018; Yin et al., 2020). Zhang and Fang (2006), Li et al. (2018), and Wang Y. et al. (2020) found significant differences in the presence of some dipeptides between thermophilic and mesothermal proteins. In addition, Gromiha et al. (1999) found that protein stability was associated with the balance between packing and solubility.

Many studies have been conducted on methods of distinguishing thermophilic proteins from normal-temperature proteins based on protein properties. Liang et al. (2005) proposed an amino acid coupling model with strong statistical ability to distinguish between thermophilic proteins and mesophilic proteins. LogitBoost Classifier and 20 features were used to distinguish thermophilic proteins by Zhang and Fang (2007) which achieved an overall classification accuracy reaching 88.9%. Montanucci et al. (2008) applied support vector machine (SVM) to investigate the impacts of mutations on the thermal stability of proteins, and with jackknife cross-validation, they achieved a prediction accuracy of 88%. Recently, Lin and Chen (2011) used feature selection technique and SVM with 30 parameters to predict thermotropic proteins, and the overall accuracy reached 93.27%. These methods have achieved good accuracy, but there remains room for improvement in the number of features used and prediction performance.

In this work, we used the data set of Lin and Chen (2011) after eliminating redundancy to distinguish between thermophilic proteins and non-thermophilic proteins. After feature extraction, MRMD2.0 was applied for feature selection and dimension reduction, and LIBSVM was used to obtain the optimal parameters of the model and establish the prediction model. Finally, from the results of cross-validation, both the number of features and the prediction accuracy were improved; the overall prediction accuracy with only 16 features in AAC was increased to 93.33%, and the highest overall accuracy, attained with 119 parameters, reached 96.02%. In addition, we analyzed the importance of features and demonstrated the strong performance of SVM by comparing this method with other methods.

MATERIALS AND METHODS

Data Sets

In this article, we conducted prediction experiments using two groups of data, namely, a group of thermophilic protein data and a group of non-thermophilic protein data. The data sets were collected by Lin and Chen (2011). Generally, thermophilic proteins and non-thermophilic proteins derive from the corresponding biosome, and optimum growth temperature is the key feature used to distinguish thermophilic and non-thermophilic proteins. Therefore, we used 60°C as the minimum optimum growth temperature for thermophilic proteins and 30°C as the maximum optimum growth temperature for non-thermophilic proteins to avoid the problem of protein denaturation. As a result, 136 prokaryotic genomes conforming to the standard were selected, and their protein sequences were obtained from the Universal Protein Resource.

Next, we screened the protein sequences to increase the quality of the data sets. The filtering process employed the following criteria: (1) the sequence must have manual annotation and evaluation; (2) the protein sequence cannot include ambiguous residue; (3) the sequences cannot be fragments of other proteins; and (4) the sequence cannot be deduced from prediction or homology. After the above screening process, we obtained a total of 1,250 non-thermophilic proteins and 1,329 thermophilic proteins. Next, highly similar sequences were removed by employing the CD-HIT program, resulting in 793 non-thermophilic proteins and 915 thermophilic proteins.

Feature Extraction

Before protein prediction, the features of the protein sequences were extracted to construct the feature vectors (**Figure 1**). For this purpose, iFeature was used, which is a utility toolkit based on python to obtain miscellaneous numerical feature representation schemes for protein sequences (Chen et al., 2018). When using iFeature, users can combine various feature clustering, feature selection, and dimension reduction algorithms to promote the analysis of feature importance and model training. iFeature has been widely tested to ensure the validity of our calculations to further ensure the strength of our work.

We used iFeature to extract the features of the protein sequences from our data set, including AAC (Bhasin and Raghava, 2004; Pan et al., 2018; Chen et al., 2019b; Liu et al., 2019; Shen et al., 2019b; Tang et al., 2019; Li Y. H. et al., 2020), C/T/D composition (CTDC), C/T/D transition (CTDT), conjoint triad (CTriad), dipeptide deviation from the expected mean (DDE) (Saravanan and Gautham, 2015), DPC (Saravanan and Gautham, 2015; Chen et al., 2019a), tripeptide composition (TPC), composition of k-spaced amino acid pairs (CKSAAP), grouped dipeptide composition (GDPC), and grouped tripeptide composition (GTPC). The following is a concise explanation of the feature extraction protocol. In all of the following formulas, n denotes the length of the protein sequence.

AAC

AAC refers to the frequency of each amino acid in a protein or peptide sequence. There are 20 kinds of naturally occurring amino acids, namely, ACDEFGHIKLMNPQRSTVWY, and their frequencies in a sequence can be calculated by the following formula:

$$f\left(i\right)=\frac{n(i)}{n},\ i\in\{A,C,D,E,F,\ldots,W,Y\}$$

where $n(i)$ refers to the number of occurrences of amino acid i.

DPC

DPC refers to the frequency of dipeptide combinations in a protein or peptide sequence, which yields 400 descriptors (Cheng J. H. et al., 2018; Tang et al., 2018). It is defined by the following formula:

$$f\left(x,y\right)=\frac{n_{xy}}{n-1},\ \ x,y\in\{A,C,D,E,F,\ldots,W,Y\}$$

where n_{xy} refers to the number of dipeptides denoted by amino acids x and y.

TPC

TPC refers to the frequency of tripeptide combinations in a protein or peptide sequence, which yields 8,000 descriptors (Tan et al., 2019; Zhu et al., 2019). It is defined by the following formula:

$$f\left(x,y,z\right)=\frac{n_{xyz}}{n-2},\ \ x,y,z\in\{A,C,D,E,F,\ldots,W,Y\}$$

where n_{xyz} refers to the number of tripeptides denoted by amino acid combination x, y, and z.

DDE

The DDE eigenvector is constructed by calculating three parameters: dipeptide composition (D_c), theoretical mean value

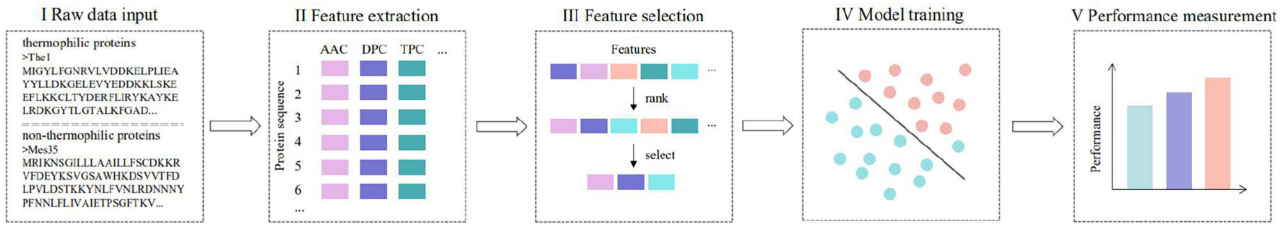


FIGURE 1 | Study flowchart. (I) The original protein sequence is input for feature extraction. (II) A feature extraction algorithm is used to obtain feature descriptors of each protein. (III) MRMD2.0 is used to rank the importance of features and select features. (IV) Support vector machine is used for parameter optimization and training model establishment. (V) Three parameters are used to evaluate the performance of the model: sensitivity (SE), specificity (SP), and accuracy (ACC).

(T_m), and theoretical variance (T_v). These three parameters and DDE are calculated as follows:

$$D_c(x,y) = \frac{n_{xy}}{n-1}, \quad x,y \in \{A, C, D, E, F, \ldots, W, Y\}$$

where n_{xy} refers to the number of dipeptides displayed by amino acid combination x and y.

$$T_m(x,y) = \frac{C_x}{C_n} \times \frac{C_y}{C_n}, \quad x,y \in \{A, C, D, E, F, \ldots, W, Y\}$$

where C_x and C_y are the number of codons encoding the first and second amino acids, respectively, in dipeptide "x, y," and C_n is the total number of possible codons remaining after removing the 3 terminated codons.

$$\begin{aligned} T_v(x,y) &= \frac{T_m(x,y)\,(1-T_m(x,y))}{n-1}, \\ &\quad x,y \in \{A, C, D, E, F, \ldots, W, Y\} \\ DDE(x,y) &= \frac{D_c(x,y) - T_m(x,y)}{\sqrt{T_v(x,y)}} \end{aligned}$$

GDPC

The GDPC encoding is a change of the DPC descriptor that includes a total of 25 descriptors, defined as follows:

$$f(x,y) = \frac{n_{xy}}{n-1}, \quad x,y \in \{g1, g2, g3, g4, g5\}$$

where n_{xy} refers to the number of dipeptides denoted by amino acid groups x and y.

GTPC

The GTPC is another change of TPC descriptor, which consists of a total of 125 descriptors and is defined as follows:

$$f(x,y,z) = \frac{n_{xyz}}{n-2}, \quad x,y,z \in \{g1, g2, g3, g4, g5\}$$

where n_{xyz} refers to the number of tripeptides denoted by amino acid combination x, y, and z.

CTD

CTD features represent the structural or physicochemical distribution patterns of amino acids in protein or peptide sequences (Dubchak et al., 1999; Tang et al., 2020). Thirteen types of physicochemical properties were used to calculate these characteristics, including hydrophobicity, standardized van der Waals volume, solvent accessibility, polarity, secondary structure, polarizability, and charge. These descriptors were computed by the following procedures: (1) the amino acid sequences were changed into residues with certain structural or physicochemical properties; (2) according to the main cluster of Tomii and Kanehisa (1996) amino acid index, the 20 amino acids were divided into 3 groups according to 7 physicochemical properties.

CTDC

After all 20 amino acids are divided into three groups, the composition descriptor is composed of 3 values, which are the total percentages of group 1, group 2, and group 3 of the protein sequences. The descriptor is calculated as follows:

$$C(x) = \frac{n(x)}{n}, \quad x \in \{\text{group 1, group 2, group 3}\}$$

where $n(x)$ refers to the number of occurrences of amino acid x in the encoded sequence.

CTDT

The transformation descriptor T also contains three values. The transition from group 1 to group 2 is the percentage frequency of a residue from group 1 followed by a residue from group 2 or a residue from group 2 followed by a residue from group 1. Transformations between group 2 and group 3 and between group 3 and group 1 are defined in a similar manner. The transformation descriptor can be calculated as follows:

$$T(x,y) = \frac{n(x,y) + n(y,x)}{n-1},$$
$$x,y \in \{(\text{group 1, group 2}), (\text{group 2, group 3}), (\text{group 3, group 1})\}$$

where $n(x,y)$ and $n(y,x)$ refer to the numbers of dipeptides denoted by "x, y" and "y, x," respectively, in the protein sequence.

Feature Selection

Feature selection is an important step in the process of protein classification (**Figure 1**) (Feng et al., 2017; Cheng, 2019; Liu, 2019; Yang W. et al., 2019; Zheng et al., 2019; Wang M. et al., 2020; Yang et al., 2020b; Zhao et al., 2020). MRMD2.0 is a very deep feature selection method, which uses the concept of the PageRank algorithm and is combined with methods such as analysis of variance (Scheffe, 1960), minimal redundancy and maximal relevance (Ding and Peng, 2005), maximal information coefficient, and least absolute shrinkage and selection operator (Xu et al., 2017). As a result, MRMD2.0 integrates seven different feature ranking algorithms with PageRank algorithm and detects optimized dimensionality with forward adding strategy. PageRank algorithm was originally used to attach weight value to each target page: pages with large weight values are displayed in the front, whereas pages with small weight values are displayed in the back. Similarly, MRMD2.0 uses PageRank algorithm and several other feature ranking algorithms to generate a corresponding weight value for each feature to form a ranking of the importance of all features.

In this study, MRMD2.0 was used to select features and reduce the dimension of the obtained features to improve the feature prediction ability. By treating each group of features in the previous step with MRMD2.0, we obtained the combination of features with the highest classification accuracy and the importance ranking of each group of features. Generally, the combination of features with the highest classification accuracy has fewer dimensions, so we refer to this process as feature dimension reduction. Based on the classification performance, we ranked the group of features. After combining the features with good classification performance, we applied MRMD2.0 to select them again. Finally, after comparing the results, we obtained the combination of features with the best classification ability.

In addition, we applied MRMD2.0 to obtain the importance ranking of features. On the rank list, higher-ranked features are more predictive; accordingly, we identified the most important features for the classification of thermophilic proteins and non-thermophilic proteins. The resulting information enhances our knowledge of the properties of proteins and can aid the construction of stable proteins in protein engineering.

LIBSVM

In this study, LIBSVM was used to construct models and make predictions (**Figure 1**). LIBSVM is an effective SVM pattern recognition and regression software package designed by Chih-Jen Lin, a professor at Taiwan University, and has been applied in many fields (Lin et al., 2012; Liu et al., 2012, 2017; Ding et al., 2017; Zeng et al., 2017; Wei et al., 2018, 2019; Xu et al., 2018b,c; Cheng et al., 2019b; Deng et al., 2019; Liang et al., 2019; Shen et al., 2019b,a; Su et al., 2019; Yang H. et al., 2019; Li F. et al., 2020; Wang H. et al., 2020; Yang et al., 2020a; Zhang et al., 2020). Before training SVM on a problem, the parameters must be specified (Jiang et al., 2013; Zhao et al., 2015, 2017). We selected the best parameters, C and g, through a simple tool provided by LIBSVM for evaluating a grid of parameters. The accuracy for each parameter setting is obtained in LIBSVM, allowing the parameters with the highest cross-validation accuracy to be determined. Next, we trained the whole data set with the best parameters C and g to obtain the prediction model. Finally, we tested and predicted our data set with the obtained model.

Performance Measurement

We used three commonly used indicators to evaluate model performance: sensitivity (SE), specificity (SP), and accuracy (ACC) (**Figure 1**) (Wang et al., 2010; Wei et al., 2017a,b; Zhang et al., 2018; Cheng et al., 2019a; Ding et al., 2019a; Junwei et al., 2019; Liang et al., 2019; Liu and Li, 2019; Tian et al., 2019; Jia et al., 2020; Liu and Chen, 2020; Li J. et al., 2020; Lv et al., 2020; Wang Z. et al., 2020). They are described as follows:

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

where TN, TP, FN, and FP refer to the numbers of correctly predicted non-thermophilic proteins, correctly predicted non-thermophilic proteins, incorrectly predicted non-thermophilic proteins, and incorrectly predicted thermophilic proteins, respectively. SE and SP indicators measure the predictive ability of a model in positive and negative situations, respectively, and ACC is used to evaluate the overall performance of a prediction model (Wang et al., 2008; Zou et al., 2017a,b; Cheng L. et al., 2018; Wang G. et al., 2018; Xue et al., 2018; Xu et al., 2018a, 2019; Ding et al., 2019b; Shen et al., 2019b; Yang, 2019; Zeng et al., 2019; Fu et al., 2020; Hong et al., 2020).

RESULTS AND DISCUSSION

Identification of Protein Thermostability

The results of feature selection by using MRMD2.0 are shown in **Table 1**. Among them, features with good classification performance include AAC, DPC, CTDC, and dipeptide deviation from the expected mean. However, although the classification ACC of dipeptide deviation from the expected mean after dimension reduction reached 85.6%, it had 365-dimensional features. Considering the excessive dimension and the unexceptional performance, only AAC, DPC, and CTDC were subsequently combined for classification.

Next, based on LIBSVM and grid parameter optimization, we used various combinations of these three features to construct models and perform cross-validation for our data sets. The results are shown in **Table 2**. The overall ACC of three schemes is higher than that of Lin and Chen (2011) (93%).

Initially, we used AAC with 16 dimensions alone to build a prediction model for the data set, achieving an overall ACC rate of 93.33% through cross-validation, which is slightly higher than that of Lin and Chen (2011). In addition, Zhang and Fang (2006) and Gromiha and Suresh (2010) used all 20 amino acids

TABLE 1 | The results of feature selection by using MRMD2.0.

Feature	Dimensions	Accuracy (%)
AAC	16/20	87.94
DPC	103/400	87.00
DDE	365/400	85.60
CTDC	33/39	85.01
CTDT	39/39	80.50
CTriad	338/343	79.80
CKSAAP	143/150	79.04
GTPC	107/125	78.63
GDPC	13/25	78.57
TPC	1,008/1023	77.11

The two numbers in the second column of the table are the number after dimension reduction and the number before dimension reduction.

TABLE 2 | The results of classification using SVM and various feature combinations.

Feature combination	SE (%)	SN (%)	Accuracy (%)
The method of Lin and Chen (2011)	93.77	92.69	93.27
AAC (16)	93.44	93.19	93.33
AAC (16) + CTDC (33)	93.77	92.81	93.33
AAC (16) + DPC (103)	95.85	96.22	96.02

The numbers in parentheses in the first column of the table represent the number of arguments to the feature preceding the parentheses.

TABLE 3 | The results of classification accuracy using LIBSVM and various combinations of important features.

Dimension	Feature	Accuracy (%)
1	K	76.41
2	K + D	77.50
3	K + D + LK	78.29

A plus sign in the second column of the table indicates the use of these characteristics for model training and classification. For example, "K + D" indicates the modeling and classification of the data sets with the two-dimension characteristics K and D.

composition to predict the thermostability of protein, and their overall ACC was 90.5 and 89%, respectively. Furthermore, Wang and Li (2014) enhanced the ACC to 95% by selecting 9 AAC and 38 DPC using a genetic algorithm. In contrast, the scheme used only 16 parameters, but the ACC reached 93.33%, which is fewer than the dimensions used in previous studies. The results show that AAC plays an important role in the identification of thermophilic proteins.

The top two features in **Table 3** were AAC and DPC. The model constructed with 16 parameters of AAC and 103 parameters of DPC achieved the highest overall ACC of 96.02%. The SE and SP of this method were 95.85 and 96.22%, respectively, which indicates that the predictive ability of this model in both positive and negative situations is excellent.

In addition, we used the combination of AAC with 16 dimensions and CTDC with 33 dimensions to build a prediction model and obtained the same overall ACC as the first model. However, this second model had higher SE and lower SP than the first model, indicating that it was slightly inferior to the model built with 16 dimensions of AAC.

Feature Importance

We aimed to identify the most important features of the method with 119 parameters that can achieve the highest ACC and analyze them. To assess feature importance, first, we used MRMD2.0 to rank all 119 features by importance. We found that the top three features were K, D, and LK (Feature K is the percentage of lysine in the amino acid sequence, feature D is the percentage of aspartic acid in the amino acid sequence, and feature LK is the percentage content of the dipeptide consisting of leucine and lysine in the amino acid sequence). These three features are arguably the most predictive among the 119 features for the classification of thermophilic proteins.

Next, to obtain the classification performance of the above features, we used one-dimensional (K), two-dimensional (K and D), and three-dimensional (K, D, and LK) features to classify our data set based on LIBSVM. The results are shown in **Table 3**.

As seen from **Table 3**, the classification ACC of the K feature alone reached 76.41%, whereas the ACC achieved with K combined with D and LK was only slightly greater. To better analyze the classification ability of these three important features, we constructed a violin diagram, scatter diagram, and 3D scatter diagram for the 1-, 2-, and 3-dimension features. The results are shown in **Figure 2**.

As seen from **Figure 2A**, the K value of the thermophilic proteome is concentrated ∼0.08, whereas the K value of the non-thermophilic proteome is concentrated ∼0.03. These results indicate that the K feature can well distinguish thermophilic proteins from non-thermophilic proteins, a finding of great significance for the identification of the thermophilic properties of proteins. All three panels reveal obvious differences in the distribution pattern between the two data sets, which indicates that these features have strong recognition ability and good performance in distinguishing thermophilic proteins from non-thermophilic proteins, as shown in **Table 3**.

Comparison With Other Classification Methods

To reveal the advantage of our method, we applied six other classification methods to train our data sets based on the Waikato environment for knowledge analysis (Weka) tool (Witten and Frank, 2002): logistic, random forest, BayesNet, logistic model trees (LMTs), J48, and reduced error pruning tree (REPTree).

We used the combination with the highest overall ACC in this article (16 features in AAC and 103 features in DPC) as the input, and we used the above classifiers to predict the data set to obtain the SE, SP, and ACC of each method. To ensure a robust comparison, we also used cross-validation to predict the data set. By comparing the performance of different methods, the performance of

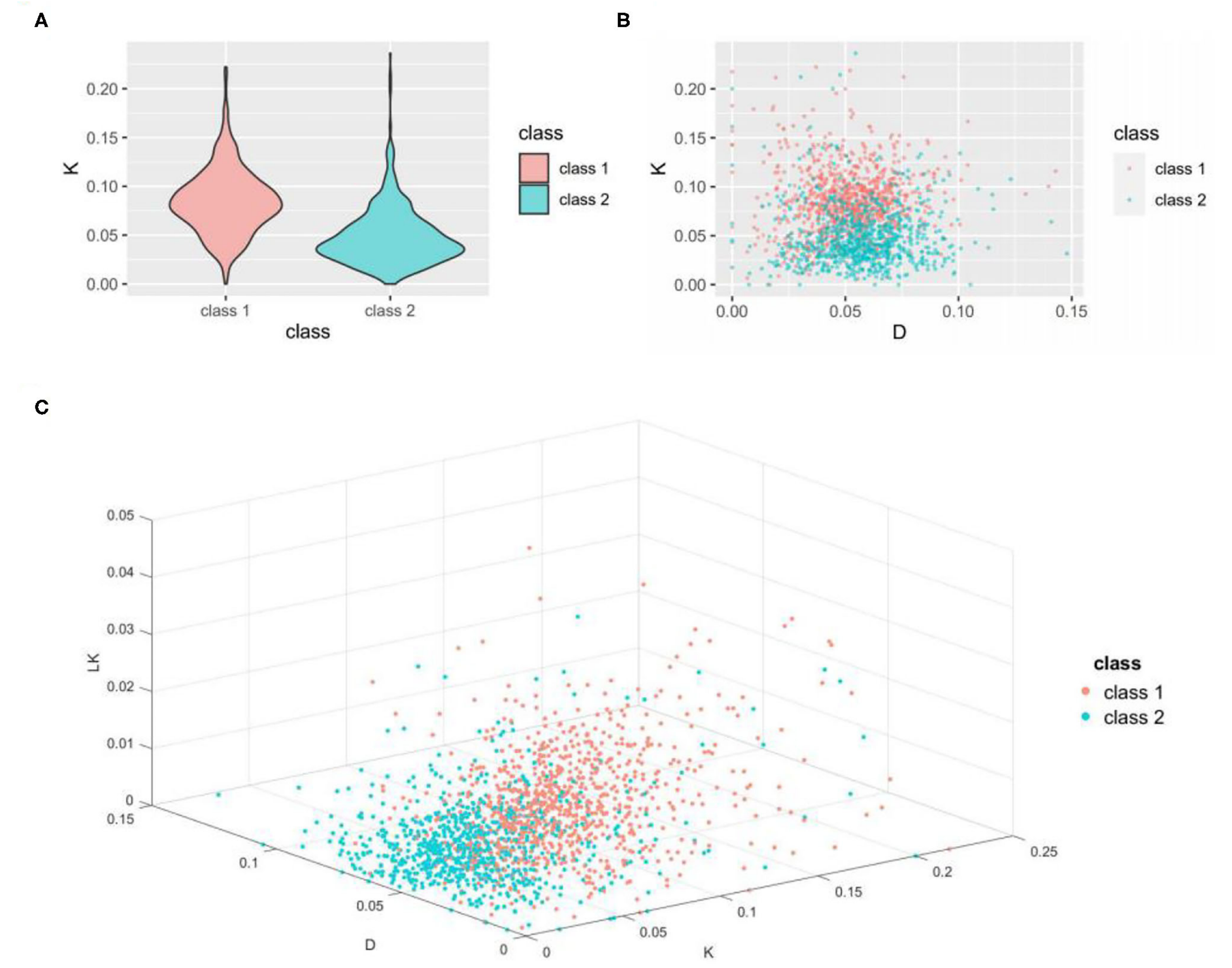


FIGURE 2 | Visualization of the ability of important features to classify thermophilic and non-thermophilic proteins. **(A)** is a violin diagram of the K feature. **(B)** is a scatter diagram of the K feature and D feature. **(C)** is a 3D scatter diagram of the K, D, and LK features. K is the percentage of lysine in the amino acid sequence, D is the percentage of aspartic acid in the amino acid sequence, and LK is the percentage content of the dipeptide consisting of leucine and lysine in the amino acid sequence.

TABLE 4 | The performance of different classification methods in the prediction of the data sets.

Classification method	SE (%)	SN (%)	Accuracy (%)
SVM (this article)	95.85	96.22	96.02
LMT	92.35	90.29	91.40
Logistic	91.15	88.90	90.11
Random Forest	91.69	87.51	89.75
BayesNet	88.08	86.25	87.24
REPTree	83.60	84.62	84.07
J48	83.50	80.33	82.03

different classifiers was evaluated. The prediction results of each method applied to the data set are shown in **Table 4**.

It can be seen from **Table 4** that the SVM we used in this study achieved the best performance; the SE, SP, and ACC of the other methods were all lower than those of the SVM method of this article. To visualize the data, we constructed a cluster histogram of the performance of the different methods, shown in **Figure 3**.

The advantage of using SVM to predict data sets is apparent from the histogram.

CONCLUSION

In this article, we distinguished 915 thermophilic proteins and 793 non-thermophilic proteins. We applied iFeature to extract the features of the protein sequences. MRMD2.0 was used to reduce the dimensions of features and select the ones that performed the best. LIBSVM was used to optimize the parameters and establish the prediction model. As a result, the overall ACC

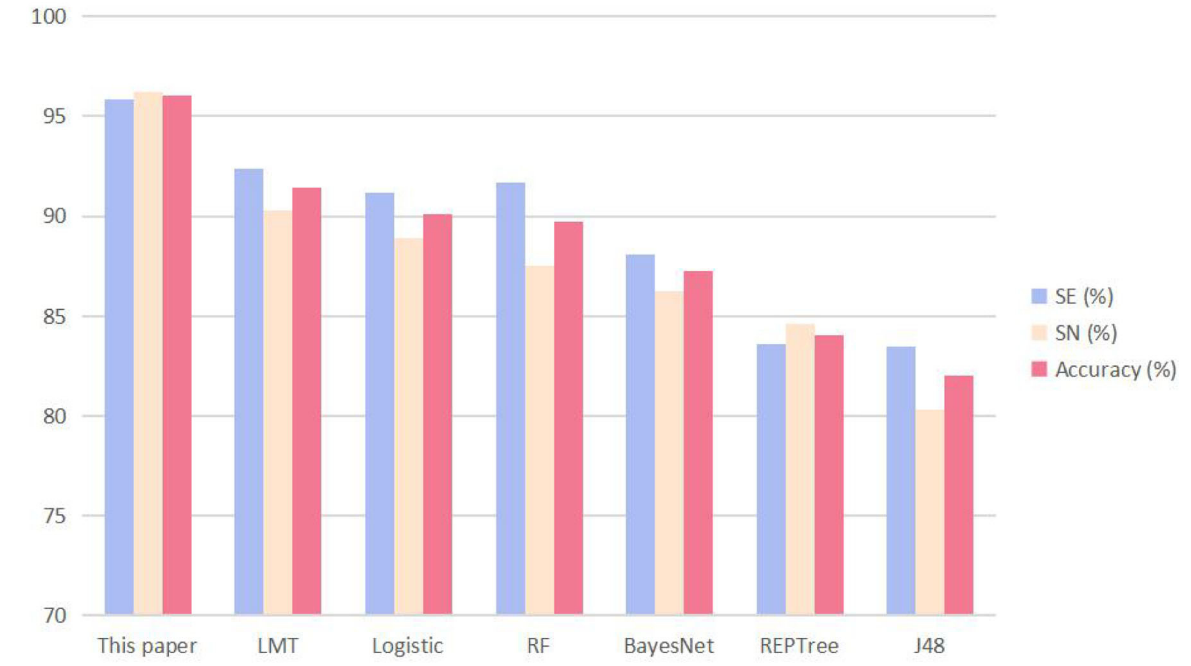


FIGURE 3 | The performance of the method described in this article and other six predictors when the input is 16 parameters of amino acid composition and 103 parameters of dipeptide composition. The performance metrics are sensitivity (SE), specificity (SP), and accuracy (ACC).

was improved, which reached 96.02% under cross-validation. Furthermore, we constructed a prediction model by LIBSVM with 16 parameters, and the ACC determined by cross-validation was 93.33%. In addition, we found that the K feature played a significant role in the identification. Finally, we demonstrated the advantage of SVM by comparing its performance with that of other methods. We aim to analyze information, such as the family of misclassified proteins, to optimize our method in the future.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: doi: 10.1016/j.mimet.2010.10.013.

AUTHOR CONTRIBUTIONS

ZG made the design of the subject and the whole idea of the whole experiment, did comparative experiments, and the analysis of the experiment. PW did experimental data analysis. ZL and YZ analyzed the results of the experiment and made some improvements to this paper. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the Natural Science Foundation of China (Nos. 61971119 and 61672328).

REFERENCES

Bhasin, M., and Raghava, G. P. S. (2004). Classification of nuclear receptors based on amino acid composition and dipeptide composition. *J. Biol. Chem.* 279, 23262–23266. doi: 10.1074/jbc.M401932200

Chen, W., Feng, P., Liu, T., and Jin, D. (2019b). Recent advances in machine learning methods for predicting heat shock proteins. *Curr. Drug Metab.* 20, 224–228. doi: 10.2174/1389200219666181031105916

Chen, W., Feng, P., and Nie, F. (2019a). *iATP*: a sequence based method for identifying anti-tubercular peptides. *Med. Chem.* 16, 620–625. doi: 10.2174/1573406415666191002152441

Chen, Z., Zhao, P., Li, F., Leier, A., Marquezlago, T. T., Wang, Y., et al. (2018). iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics* 34, 2499–2502. doi: 10.1093/bioinformatics/bty140

Cheng, J. H., Yang, H., Liu, M. L., Su, W., Feng, P. M., Ding, H., et al. (2018). Prediction of bacteriophage proteins located in the host cell using hybrid features. *Chemometr. Intell. Lab.* 180, 64–69. doi: 10.1016/j.chemolab.2018.07.006

Cheng, L. (2019). Computational and biological methods for gene therapy. *Curr. Gene Ther.* 19, 210–210. doi: 10.2174/156652321904191022113307

Cheng, L., Jiang, Y., Ju, H., Sun, J., Peng, J., Zhou, M., et al. (2018). InfAcrOnt: calculating cross-ontology term similarities using information flow by a random walk. *BMC Genomics* 19(Suppl. 1):919. doi: 10.1186/s12864-017-4338-6

Cheng, L., Wang, P., Tian, R., Wang, S., Guo, Q., Luo, M., et al. (2019a). LncRNA2Target v2.0: a comprehensive database for target genes of lncRNAs in human and mouse. *Nucleic Acids Res.* 47, D140–D144. doi: 10.1093/nar/gky1051

Cheng, L., Zhuang, H., Ju, H., Yang, S., Han, J., Tan, R., et al. (2019b). Exposing the causal effect of body mass index on the risk of type 2

Diabetes mellitus: a mendelian randomization study. *Front. Genet*. 10:94. doi: 10.3389/fgene.2019.00094

Deng, L., Wang, J., and Zhang, J. (2019). Predicting gene ontology function of human MicroRNAs by integrating multiple networks. *Front. Genet*. 10:3. doi: 10.3389/fgene.2019.00003

Ding, C., and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *J. Bioinform. Comput. Biol*. 3, 185–205. doi: 10.1142/S0219720005001004

Ding, Y., Tang, J., and Guo, F. (2017). Identification of drug-target interactions via multiple information integration. *Inf. Sci.* 418–419, 546–560. doi: 10.1016/j.ins.2017.08.045

Ding, Y., Tang, J., and Guo, F. (2019a). Identification of drug-side effect association via multiple information integration with centered kernel alignment. *Neurocomputing* 325, 211–224. doi: 10.1016/j.neucom.2018.10.028

Ding, Y., Tang, J., and Guo, F. (2019b). Identification of drug-side effect association via semi-supervised model and multiple kernel learning. *IEEE J. Biomed. Health Inform.* 23, 2619–2632. doi: 10.1109/JBHI.2018.2883834

Dubchak, I., Muchnik, I., Mayor, C., Dralyuk, I., and Kim, S. (1999). Recognition of a protein fold in the context of the SCOP classification. *Proteins* 35, 401–407.

Feng, P., Ding, H., Lin, H., and Chen, W. (2017). AOD: the antioxidant protein database. *Sci. Rep*. 7:7449. doi: 10.1038/s41598-017-08115-6

Fu, X., Cai, L., Zeng, X., and Zou, Q. (2020). StackCPPred: a stacking and pairwise energy content-based prediction of cell-penetrating peptides and their uptake efficiency. *Bioinformatics* 36, 3028–3034. doi: 10.1093/bioinformatics/btaa131

Gromiha, M. M., Oobatake, M., and Sarai, A. (1999). Important amino acid properties for enhanced thermostability from mesophilic to thermophilic proteins. *Biophys. Chem*. 82, 51–67. doi: 10.1016/S0301-4622(99)00103-9

Gromiha, M. M., and Suresh, M. X. (2010). Discrimination of mesophilic and thermophilic proteins using machine learning algorithms. *Proteins* 70, 1274–1279. doi: 10.1002/prot.21616

Hong, Z., Zeng, X., Wei, L., and Liu, X. (2020). Identifying enhancer–promoter interactions with neural network based on pre-trained DNA vectors and attention mechanism. *Bioinformatics* 36, 1037–1043. doi: 10.1093/bioinformatics/btz694

Jia, C., Bi, Y., Chen, J., Leier, A., Li, F., and Song, J. (2020). PASSION: an ensemble neural network approach for identifying the binding sites of RBPs on circRNAs. *Bioinformatics* 36, 4276–4282. doi: 10.1093/bioinformatics/btaa522

Jiang, Q. H., Wang, G. H., Jin, S. L., Li, Y., and Wang, Y. D. (2013). Predicting human microRNA-disease associations based on support vector machine. *Int. J. Data Min. Bioinform*. 8, 282–293. doi: 10.1504/IJDMB.2013.056078

Junwei, H., Xudong, H., Qingfei, K., and Liang, C. (2019). psSubpathway: a software package for flexible identification of phenotype-specific subpathways in cancer progression. *Bioinformatics* 36, 2303–2305. doi: 10.1093/bioinformatics/btz894

Kumar, S., Tsai, C., and Nussinov, R. (2000). Factors enhancing protein thermostability. *Protein Eng*. 13, 179–191. doi: 10.1093/protein/13.3.179

Li, F., Zhou, Y., Zhang, X., Tang, J., Yang, Q., Zhang, Y., et al. (2020). SSizer: determining the sample sufficiency for comparative biological study. *J. Mol. Biol*. 432, 3411–3421. doi: 10.1016/j.jmb.2020.01.027

Li, J., Pu, Y., Tang, J., Zou, Q., and Guo, F. (2020). DeepAVP: a dual-channel deep neural network for identifying variable-length antiviral peptides. *IEEE J. Biomed. Health Inform*. doi: 10.1109/JBHI.2020.2977091. [Epub ahead of print].

Li, Y. H., Li, X. X., Hong, J. J., Wang, Y. X., Fu, J. B., Yang, H., et al. (2020). Clinical trials, progression-speed differentiating features and swiftness rule of the innovative targets of first-in-class drugs. *Brief. Bioinform*. 21, 649–662. doi: 10.1093/bib/bby130

Li, Y. H., Yu, C. Y., Li, X. X., Zhang, P., Tang, J., Yang, Q., et al. (2018). Therapeutic target database update 2018: enriched resource for facilitating bench-to-clinic research of targeted therapeutics. *Nucleic Acids Res*. 46, D1121–D1127. doi: 10.1093/nar/gkx1076

Liang, C., Changlu, Q., He, Z., Tongze, F., and Xue, Z. (2019). gutMDisorder: a comprehensive database for dysbiosis of the gut microbiota in disorders and interventions. *Nucleic Acids Res.* 48, D554–D560. doi: 10.1093/nar/gkz843

Liang, H., Huang, C., Ko, M., and Hwang, J. (2005). Amino acid coupling patterns in thermophilic proteins. *Proteins* 59, 58–63. doi: 10.1002/prot.20386

Lin, H., and Chen, W. (2011). Prediction of thermophilic proteins using feature selection technique. *J. Microbiol. Methods* 84, 67–70. doi: 10.1016/j.mimet.2010.10.013

Lin, H., Ding, C., Song, Q., Yang, P., Ding, H., Deng, K. J., et al. (2012). The prediction of protein structural class using averaged chemical shifts. *J. Biomol. Struct. Dyn*. 29, 643–649. doi: 10.1080/07391102.2011.672628

Liu, B. (2019). BioSeq-Analysis: a platform for DNA, RNA, and protein sequence analysis based on machine learning approaches. *Brief. Bioinform*. 20, 1280–1294. doi: 10.1093/bib/bbx165

Liu, B., Gao, X., and Zhang, H. (2019). BioSeq-Analysis2.0: an updated platform for analyzing DNA, RNA, and protein sequences at sequence level and residue level based on machine learning approaches. *Nucleic Acids Res*. 47:e127. doi: 10.1093/nar/gkz740

Liu, B., Li, C., and Yan, K. (2012). DeepSVM-fold: protein fold recognition by combining support vector Machines and pairwise sequence similarity scores generated by deep learning networks. *Brief. Bioinform.* 21, 1733–1741. doi: 10.1093/bib/bbz098

Liu, B., and Li, K. (2019). iPromoter-2L2.0: identifying promoters and their types by combining smoothing cutting window algorithm and sequence-based features. *Mol. Ther. Nucleic Acids*. 18, 80–87. doi: 10.1016/j.omtn.2019.08.008

Liu, K., and Chen, W. (2020). iMRM:a platform for simultaneously identifying multiple kinds of RNA modifications. *Bioinformatics* 36, 3336–3342. doi: 10.1093/bioinformatics/btaa155

Liu, Y., Zeng, X., He, Z., and Zou, Q. (2017). Inferring MicroRNA-disease associations by random walk on a heterogeneous network with multiple data sources. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 14, 905–915. doi: 10.1109/TCBB.2016.2550432

Lv, H., Dao, F. Y., Zhang, D., Guan, Z. X., Yang, H., Su, W., et al. (2020). iDNA-MS. an integrated computational tool for detecting DNA modification sites in multiple genomes. *iScience* 23:100991. doi: 10.1016/j.isci.2020.100991

Montanucci, L., Fariselli, P., Martelli, P. L., and Casadio, R. (2008). Predicting protein thermostability changes from sequence upon multiple mutations. *Bioinformatics* 2008, 190–195. doi: 10.1093/bioinformatics/btn166

Pan, Y., Wang, Z., Zhan, W., and Deng, L. (2018). Computational identification of binding energy hot spots in protein-RNA complexes using an ensemble approach. *Bioinformatics* 34, 1473–1480. doi: 10.1093/bioinformatics/btx822

Sadeghi, M., Naderimanesh, H., Zarrabi, M., and Ranjbar, B. (2006). Effective factors in thermostability of thermophilic proteins. *Biophys. Chem*. 119, 256–270. doi: 10.1016/j.bpc.2005.09.018

Saravanan, V., and Gautham, N. (2015). Harnessing computational biology for exact linear B-cell epitope prediction: a novel amino acid composition-based feature descriptor. *OMICS* 19, 648–658. doi: 10.1089/omi.2015.0095

Scheffe, H. (1960). The analysis of variance. *Soil Sci*. 89:360. doi: 10.1097/00010694-196006000-00016

Shen, C., Jiang, L., Ding, Y., Tang, J., and Guo, F. (2019b). LPI-KTASLP. prediction of lncRNA-protein interaction by semi-supervised link learning with multivariate information. *IEEE Access* 7, 13486–13496. doi: 10.1109/ACCESS.2019.2894225

Shen, Y., Ding, Y., Tang, J., Zou, Q., and Guo, F. (2019a). Critical evaluation of web-based prediction tools for human protein subcellular localization. *Brief. Bioinform.* 21, 1628–1640. doi: 10.1093/bib/bbz106

Shen, Y., Tang, J., and Guo, F. (2019b). Identification of protein subcellular localization via integrating evolutionary and physicochemical information into Chou's general PseAAC. *J. Theor. Biol*. 462, 230–239. doi: 10.1016/j.jtbi.2018.11.012

Su, R., Wu, H., Xu, B., Liu, X., and Wei, L. (2019). Developing a multi-dose computational model for drug-induced hepatotoxicity prediction based on toxicogenomics data. *IEEE/ACM Trans. Comp. Biol. Bioinform*. 16, 1231–1239. doi: 10.1109/TCBB.2018.2858756

Tan, J. X., Li, S. H., Zhang, Z. M., Chen, C. X., Chen, W., Tang, H., et al. (2019). Identification of hormone binding proteins based on machine learning methods. *Math. Biosci. Eng*. 16, 2466–2480. doi: 10.3934/mbe.2019123

Tang, H., Zhao, Y. W., Zou, P., Zhang, C. M., Chen, R., Huang, P., et al. (2018). HBPred: a tool to identify growth hormone-binding proteins. *Int. J. Biol. Sci*. 14, 957–964. doi: 10.7150/ijbs.24174

Tang, J., Fu, J., Wang, Y., Li, B., Li, Y., Yang, Q., et al. (2020). ANPELA: analysis and performance assessment of the label-free quantification workflow for metaproteomic studies. *Brief. Bioinform*. 21, 621–636. doi: 10.1093/bib/bby127

Tang, J., Fu, J., Wang, Y., Luo, Y., Yang, Q., Li, B., et al. (2019). Simultaneous improvement in the precision, accuracy, and robustness of label-free proteome quantification by optimizing data manipulation chains. *Mol. Cell. Proteomics* 18, 1683–1699. doi: 10.1074/mcp.RA118.001169

Tian, B., Wu, X., Chen, C., Qiu, W., Ma, Q., and Yu, B. (2019). Predicting protein–protein interactions by fusing various Chou's pseudo components and using wavelet denoising approach. *J. Theor. Biol.* 462, 329–346. doi: 10.1016/j.jtbi.2018.11.011

Tomii, K., and Kanehisa, M. (1996). Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins. *Protein Eng.* 9, 27–36. doi: 10.1093/protein/9.1.27

Wang, G., Luo, X., Wang, J., Wan, J., Xia, S., Zhu, H., et al. (2018). MeDReaders: a database for transcription factors that bind to methylated DNA. *Nucleic Acids Res.* 46, D146–D151. doi: 10.1093/nar/gkx1096

Wang, G., Wang, Y., Feng, W., Wang, X., Yang, J. Y., Zhao, Y., et al. (2008). Transcription factor and microRNA regulation in androgen-dependent and -independent prostate cancer cells. *BMC Genomics*. 9(Suppl. 2):S22. doi: 10.1186/1471-2164-9-S2-S22

Wang, G., Wang, Y., Teng, M., Zhang, D., Li, L., and Liu, Y. (2010). Signal transducers and activators of transcription-1 (STAT1) regulates microRNA transcription in interferon gamma-stimulated HeLa cells. *PLoS ONE* 5:e11794. doi: 10.1371/journal.pone.0011794

Wang, H., Ding, Y., Tang, J., and Guo, F. (2020). Identification of membrane protein types via multivariate information fusion with Hilbert-schmidt independence criterion. *Neurocomputing* 383, 257–269. doi: 10.1016/j.neucom.2019.11.103

Wang, H., Liu, C., and Deng, L. (2018). Enhanced prediction of hot spots at protein-protein interfaces using extreme gradient boosting. *Sci. Rep.* 8:14285. doi: 10.1038/s41598-018-32511-1

Wang, L. Q., and Li, C. F. (2014). Optimal subset selection of primary sequence features using the genetic algorithm for thermophilic proteins identification. *Biotechnol. Lett.* 36, 1963–1969. doi: 10.1007/s10529-014-1577-3

Wang, M., Yue, L., Cui, X., Chen, C., Zhou, H., Ma, Q., et al. (2020). Prediction of extracellular matrix proteins by fusing multiple feature information, elastic net, and random forest algorithm. *Mathematics* 8:169. doi: 10.3390/math8020169

Wang, Y., Zhang, S., Li, F., Zhou, Y., Zhang, Y., Wang, Z., et al. (2020). Therapeutic target database 2020: enriched resource for facilitating research and early development of targeted therapeutics. *Nucleic Acids Res.* 48, D1031–D1041. doi: 10.1093/nar/gkz981

Wang, Z., He, W., Tang, J., and Guo, F. (2020). Identification of highest-affinity binding sites of yeast transcription factor families. *J. Chem. Inf. Model.* 60, 1876–1883. doi: 10.1021/acs.jcim.9b01012

Wei, L., Wan, S., Guo, J., and Wong, K. K. (2017a). A novel hierarchical selective ensemble classifier with bioinformatics application. *Artif. Intell. Med.* 83, 82–90. doi: 10.1016/j.artmed.2017.02.005

Wei, L., Xing, P., Shi, G., Ji, Z., and Zou, Q. (2019). Fast prediction of methylation sites using sequence-based feature selection technique. *IEEE/ACM Trans. Comp. Biol. Bioinform.* 16, 1264–1273. doi: 10.1109/TCBB.2017.2670558

Wei, L., Xing, P., Zeng, J., Chen, J., Su, R., and Guo, F. (2017b). Improved prediction of protein–protein interactions using novel negative samples, features, and an ensemble classifier. *Artif. Intell. Med.* 83, 67–74. doi: 10.1016/j.artmed.2017.03.001

Wei, L., Zhou, C., Chen, H., Song, J., and Su, R. (2018). ACPred-FL: a sequence-based predictor based on effective feature representation to improve the prediction of anti-cancer peptides. *Bioinformatics* 34, 4007–4016. doi: 10.1093/bioinformatics/bty451

Witten, I. H., and Frank, E. (2002). Data mining: practical machine learning tools and techniques with Java implementations. *SIGMOD Rec.* 31, 76–77. doi: 10.1145/507338.507355

Witten, I. H., and Frank, E. (2002). Data mining: practical machine learning tools and techniques with Java implementations. *SIGMOD Rec.* 31, 76-77. doi: 10.1145/507338.507355

Xu, L., Liang, G., Liao, C., Chen, G.-D., and Chang C-C. (2019). k-skip-n-gram-RF: a random forest based method for Alzheimer's disease protein identification . *Front. Genet.* 10:33. doi: 10.3389/fgene.2019.00033

Xu, L., Liang, G., Liao, C., Chen, G.-D., and Chang, C.-C. (2018a). An efficient classifier for alzheimer's disease genes identification. *Molecules* 23:3140. doi: 10.3390/molecules23123140

Xu, L., Liang, G., Shi, S., and Liao, C. (2018b). SeqSVM. a sequence-based support vector machine method for identifying antioxidant proteins. *Int. J. Mol. Sci.* 19:1773. doi: 10.3390/ijms19061773

Xu, L., Liang, G., Wang, L., and Liao, C. (2018c). A novel hybrid sequence-based model for identifying anticancer peptides. *Genes* 9:158. doi: 10.3390/genes9030158

Xu, Y., Wang, Y., Luo, J., Zhao, W., and Zhou, X. (2017). Deep learning of the splicing (epi) genetic code reveals a novel candidate mechanism linking histone modifications to ESC fate decision. *Nucleic Acids Res.* 45, 12100–12112. doi: 10.1093/nar/gkx870

Xue, W., Yang, F., Wang, P., Zheng, G., Chen, Y., Yao, X., et al. (2018). What contributes to serotonin-norepinephrine reuptake inhibitors' dual-targeting mechanism? The key role of transmembrane domain 6 in human serotonin and norepinephrine transporters revealed by molecular dynamics simulation. *ACS Chem. Neurosci.* 9, 1128–1140. doi: 10.1021/acschemneuro.7b00490

Yang, C. (2019). Interaction of cell and gene therapy with the immune system. *Curr. Gene Ther.* 19, 69–70. doi: 10.2174/156652321902190722112944

Yang, H., Yang, W., Dao, F. Y., Lv, H., Ding, H., Chen, W., et al. (2019). A comparison and assessment of computational method for identifying recombination hotspots in *Saccharomyces cerevisiae*. *Brief. Bioinform.* 21, 1568–1580. doi: 10.1093/bib/bbz123

Yang, Q., Li, B., Tang, J., Cui, X., Wang, Y., Li, X., et al. (2020a). Consistent gene signature of schizophrenia identified by a novel feature selection strategy from comprehensive sets of transcriptomic data. *Brief. Bioinform.* 21, 1058–1068. doi: 10.1093/bib/bbz049

Yang, Q., Wang, Y., Zhang, Y., Li, F., Xia, W., Zhou, Y., et al. (2020b). NOREVA: enhanced normalization and evaluation of time-course and multi-class metabolomic data. *Nucleic Acids Res.* 48, W436–W448. doi: 10.1093/nar/gkaa258

Yang, W., Zhu, X. J., Huang, J., Ding, H., and Lin, H. (2019). A brief survey of machine learning methods in protein sub-Golgi localization. *Curr. Bioinform.* 14, 234–240. doi: 10.2174/1574893613666181113131415

Yin, J., Sun, W., Li, F., Hong, J., Li, X., Zhou, Y., et al. (2020). VARIDT 1.0: variability of drug transporter database. *Nucleic Acids Res.* 48, D1042–D1050. doi: 10.1093/nar/gkz779

Zeng, X., Liao, Y., Liu, Y., and Zou, Q. (2017). Prediction and validation of disease genes using hetesim scores. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 14, 687–695. doi: 10.1109/TCBB.2016.2520947

Zeng, X., Zhu, S., Liu, X., Zhou, Y., Nussinov, R., and Cheng, F. (2019). deepDR: a network-based deep learning approach to in silico drug repositioning. *Bioinformatics* 35, 5191–5198. doi: 10.1093/bioinformatics/btz418

Zhang, F., Ma, A., Wang, Z., Ma, Q., Liu, B., Huang, L., et al. (2018). A central edge selection based overlapping community detection algorithm for the detection of overlapping structures in protein–protein interaction *networks*. *Molecules* 23:2633. doi: 10.3390/molecules23102633

Zhang, G., and Fang, B. (2006). Discrimination of thermophilic and mesophilic proteins via pattern recognition methods. *Process Biochem.* 41, 552–556. doi: 10.1016/j.procbio.2005.09.003

Zhang, G., and Fang, B. (2007). LogitBoost classifier for discriminating thermophilic and mesophilic proteins. *J. Biotechnol.* 127, 417–424. doi: 10.1016/j.jbiotec.2006.07.020

Zhang, Z. Y., Yang, Y. H., Ding, H., Wang, D., Chen, W., and Lin, H. (2020). Design powerful predictor for mRNA subcellular location prediction in Homo sapiens. *Brief. Bioinform.* doi: 10.1093/bib/bbz177. [Epub ahead of print].

Zhao, X., Jiao, Q., Li, H., Wu, Y., Wang, H., Huang, S., et al. (2020). ECFS-DEA: an ensemble classifier-based feature selection for differential expression analysis on expression profiles. *BMC Bioinform.* 21:43. doi: 10.1186/s12859-020-3388-y

Zhao, Y., Wang, F., Chen, S., Wan, J., and Wang, G. (2017). Methods of MicroRNA promoter prediction and transcription factor mediated regulatory network. *Biomed. Res. Int.* 2017:7049406. doi: 10.1155/2017/7049406

Zhao, Y., Wang, F., and Juan, L. (2015). MicroRNA promoter identification in arabidopsis using multiple histone markers. *Biomed. Res. Int.* 2015:861402. doi: 10.1155/2015/861402

Zheng, N., Wang, K., Zhan, W., and Deng, L. (2019). Targeting virus-host protein interactions: feature extraction and machine learning approaches.

Curr. Drug Metab. 20, 177–184. doi: 10.2174/1389200219666180829121038

Zhu, X. J., Feng, C. Q., Lai, H. Y., Chen, W., and Lin, H. (2019). Predicting protein structural classes for low-similarity sequences by evaluating different features. *Knowl-Based Syst*. 163, 787–793. doi: 10.1016/j.knosys.2018.10.007

Zou, Q., Chen, L., Huang, T., Zhang, Z., and Xu, Y. (2017a). Machine learning and graph analytics in computational biomedicine. Artificial intelligence in medicine. *Artif. Intell. Med.* 83:1. doi: 10.1016/j.artmed.2017.09.003

Zou, Q., Mrozek, D., Ma, Q., and Xu, Y. (2017b). Scalable data mining algorithms in computational biology and biomedicine. *Biomed Res. Int*. 2017:5652041. doi: 10.1155/2017/5652041

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers
in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 06 January 2021
doi: 10.3389/fbioe.2020.627335

Predicting Cell Wall Lytic Enzymes Using Combined Features

*Xiao-Yang Jing and Feng-Min Li**

College of Science, Inner Mongolia Agricultural University, Hohhot, China

Due to the overuse of antibiotics, people are worried that existing antibiotics will become ineffective against pathogens with the rapid rise of antibiotic-resistant strains. The use of cell wall lytic enzymes to destroy bacteria has become a viable alternative to avoid the crisis of antimicrobial resistance. In this paper, an improved method for cell wall lytic enzymes prediction was proposed and the amino acid composition (AAC), the dipeptide composition (DC), the position-specific score matrix auto-covariance (PSSM-AC), and the auto-covariance average chemical shift (acACS) were selected to predict the cell wall lytic enzymes with support vector machine (SVM). In order to overcome the imbalanced data classification problems and remove redundant or irrelevant features, the synthetic minority over-sampling technique (SMOTE) was used to balance the dataset. The F-score was used to select features. The S_n, S_p, MCC, and Acc were 99.35%, 99.02%, 0.98, and 99.19% with jackknife test using the optimized combination feature AAC+DC+acACS+PSSM-AC. The S_n, S_p, MCC, and Acc of cell wall lytic enzymes in our predictive model were higher than those in existing methods. This improved method may be helpful for protein function prediction.

Keywords: cell wall lytic enzymes, optimized combination feature, synthetic minority over-sampling technique, F-score, support vector machine, jackknife test

Edited by:
Zhibin Lv,
University of Electronic Science and Technology of China, China

Reviewed by:
Fei Guo,
Tianjin University, China
Hua Tang,
Southwest Medical University, China

****Correspondence:***
Feng-Min Li
fmli@imau.edu.cn

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *09 November 2020*
Accepted: *04 December 2020*
Published: *06 January 2021*

Citation:
Jing X-Y and Li F-M (2021) Predicting Cell Wall Lytic Enzymes Using Combined Features. Front. Bioeng. Biotechnol. 8:627335. doi: 10.3389/fbioe.2020.627335

INTRODUCTION

Bacteria are constantly around us, and bacterial infections have become a major public health problem. The overuse of antibiotics leads to the rapid rise of antibiotic-resistant strains, and people are worried that existing antibiotics will become ineffective against pathogens. Using cell wall lytic enzymes to destroy bacteria has become a viable alternative method to avoid the crisis of antimicrobial resistance (Sommer et al., 2017; Wu et al., 2017; Bhagwat et al., 2019; Cheng et al., 2020). Cell wall lytic enzymes are divided into two enzymes: endolysin and autolysin. Endolysins are phage-encoded enzymes that have evolved to degrade the bacterial cell wall (Shavrina et al., 2016). Many studies have shown that endolysin has an excellent bactericidal effect on *Staphylococcus aureus* (Ajuebor et al., 2016), *Escherichia coli* (Yan et al., 2019), *Streptococcus suis* (Der Ploeg, 2008), and other pathogens. Compared with conventional antibiotics, endolysin has many advantages, such as rapid host killing, host specificity, low chances of developing drug resistance, and efficacy against multidrug-resistant bacteria (Gondil et al., 2020). Autolysin is the other cell wall lytic enzyme that degrades some bonds in the peptidoglycan backbone of the bacterial cell wall (Usobiaga et al., 1996), and it is closely related to the life of cells and participates in the control of cell growth, cell lysis, daughter-cell separation, and biofilm formation (Kalali et al., 2019). Cell wall lytic enzymes have become a valuable tool for biological researchers in the medical and food industry and in agricultural applications (Yu, 1997).

Experimental determination of the cell wall lytic enzymes is time-consuming and laborious, so it is necessary to use an effective method to predict cell wall lytic enzymes. Recently some computational methods for predicting cell wall lytic enzymes have been proposed. Ding et al. (2009) used Chou's amphiphilic pseudo to predict cell wall lytic enzymes; the predictive accuracy was 80.40% with jackknife test. Chen et al. (2016) developed a predictor called "Lypred" that used pseudo amino acid composition (PseAAC) as a feature vector; the predictive accuracy was 91.3% with fivefold cross-validation. Meng et al. (2020) developed a predictor called "CWLy-SVM" that employed the 473-dimensional sequence-based feature descriptor to predict cell wall lytic enzymes; the result was 95.50% with jackknife test. In this paper, the amino acid composition (AAC), the dipeptide composition (DC), the position-specific score matrix auto-covariance (PSSM-AC), and the Auto-covariance average chemical shift (acACS) were used to predict the cell wall lytic enzymes with the same datasets as investigated by Chen et al. (2016).

Data imbalance is always considered a problem in developing efficient and reliable prediction systems; in imbalanced datasets, the classifier would tend to the majority class. Here, the synthetic minority over-sampling technique (SMOTE) was used to solve the problem of imbalance. To remove redundant or irrelevant features, we selected features using the F-score algorithm. The accuracy (Acc) was 99.19% with a balanced dataset in jackknife test by using the optimized combination feature AAC+DC+PSSM-AC+acACS.

MATERIALS AND METHODS

Benchmark Dataset

The benchmark dataset was generated by Chen et al. (2016), The dataset was taken from the Universal Protein Resource (UniProt), using the following steps to collect the sequence: (1) sequences annotated with "Inferred from homology" or "Predicted" were removed. (2) Sequences which were the fragments of other proteins were not included. (3) Sequences containing ambiguous letters such as "B," "J," "O," "U," "X," and "Z" were excluded. To reduce homologous bias and redundancy, the program CD–HIT (Li and Godzik, 2006) was used to remove those sequences that have $\geq$ 40% pairwise sequence identity. Finally, 375 sequences were obtained; they contained 68 lyases and 307 non-lyases, and the dataset can be expressed as:

$$S = S_{lysases} \cup S_{nonlysases} \tag{1}$$

The dataset can be freely downloaded from http://lin-group.cn/server/Lypred/data.html.

Feature Extraction Techniques

Feature extraction is a crucial step in developing a powerful predictor; a set of reasonable features contains more protein sequence information (Zhu et al., 2018; Yang et al., 2019; Zhang and Liu, 2019). Generally, the feature combination can boost the prediction performance. In this paper, the AAC, the DC, the PSSM-AC, and the acACS were used to predict the cell wall lytic enzymes.

Amino Acid Composition

The amino acid composition of proteins is the most basic feature information in all features. The protein sequence consists of 20 amino acids (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, and Y). AAC calculates the occurrence frequency of the 20 native amino acids so that the protein sequence can be expressed as 20 features in a feature vector. It can be defined as:

$$P = [x_1, x_2, x_3, \cdots, x_i, \cdots, x_{20}] \tag{2}$$

$$x_i = \frac{n_i}{L} \tag{3}$$

Where n_i is the occurrence number of the 20 native amino acid in protein sequence and L is the length of the protein sequence.

Dipeptide Composition

Dipeptide composition (DC) is calculated as the occurrence frequency of each two adjacent amino acid residues. There are 20*20 = 400 combinations of amino acid pairs. Compared with AAC, DC is a feature that considers some sequence-order information. It can be calculated as:

$$P = \left[f_1, f_2, f_3, \ldots, f_i, \ldots, f_{400}\right] \tag{4}$$

$$f_i = \frac{m_i}{L-1} \tag{5}$$

Where m_i is the occurrence number of i-th dipeptide in protein sequence and L is the length of the protein sequence.

Position-Specific Score Matrix Auto-Covariance

Position-Specific Score Matrix Auto-Covariance (PSSM-AC) is a feature that extracts the evolutionary information of a protein sequence. PSSM-AC was first proposed to predict the protein fold recognition by Dong et al. (2009). Recently, the PSSM-AC was used successfully in many works for the prediction of protein function (Zou et al., 2013; Huang and Li, 2018; Wang et al., 2019b, 2020a). In PSSM-AC, the PSI-BLAST (Position-Specific Iterative Basic Local Alignment Tool) was used to generate PSSM; the threshold of e-value is 0.001 and the maximum number of iterations is 3. PSSM-AC is calculated as the correlation between two residues within PSSM. This method can be represented as:

$$P_{PSSM} = \begin{bmatrix} R_{1,1} & R_{1,2} & \ldots & R_{1,j} & \ldots & R_{1,20} \\ R_{2,1} & R_{2,2} & \ldots & R_{2,j} & \ldots & R_{2,20} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{i,1} & R_{i,2} & \ldots & R_{i,j} & \ldots & R_{i,20} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{L,1} & R_{L,2} & \ldots & R_{L,j} & \ldots & R_{L,20} \end{bmatrix} \tag{6}$$

$$P_{PSSM} - AC\left(j, \text{lg}\right) = \frac{1}{L - \text{lg}} \sum_{i=1}^{L-\text{lg}} \left(R_{i,j} - \overline{R}_j\right)\left(R_{i+\text{lg},j} - \overline{R}_j\right) \tag{7}$$

$$\overline{R}_j = \frac{1}{L}\sum_{i=1}^{L} R_{i,j}\left(j = 1, \ldots, 20\right) \quad (8)$$

Where $R_{i,j}$ is the score of the residue of the i-th position mutated to the j-th amino acids residue in the protein sequence; a high score means a highly conserved position. L is the length of the protein sequence, lg is the distance along the sequence, and $0 < lg < L$. As a result, the protein sequence generates a 20 × lg dimensional feature vector with PSSM-AC.

Auto-Covariance Average Chemical Shift

As important parameters are measured by nuclear magnetic resonance (NMR) spectroscopy, the chemical shift has been used as a powerful indicator of the protein structure. Several researchers revealed that the average chemical shift (ACS) of a particular nucleus in the protein backbone empirically correlates to its secondary structure (Sibley et al., 2003). acACS was proposed by Fan et al. (2014), In acACS, the secondary structure was converted into the average chemical shift, and then the auto-covariance function was used to construct the vector representing the protein sequence by selecting different. In this work, the secondary structure was obtained by submitting the protein sequence to PSIPRED[1], and then the protein sequence and the corresponding secondary structure were submitted to the acACS web server[2]. It can be calculated as:

For a protein P, where each amino acid in the sequence is substituted by its averaged chemical shift, P can be expressed as:

$$P = \left[A_1^i, A_2^i, A_3^i, \ldots, A_L^i\right]\left(i = {}^{15}N, {}^{13}C_\alpha, {}^{1}H_\alpha, {}^{1}H_N\right) \quad (9)$$

Where ${}^{15}N$ stands for Nitrogen, ${}^{13}C_\alpha$ for alpha Carbon, ${}^{1}H_\alpha$ for alpha Hydrogen, and ${}^{1}H_N$ for Hydrogen linked with Nitrogen.

After we select λ = 17 and $i = {}^{15}N, {}^{13}C_\alpha, {}^{1}H_\alpha, {}^{1}H$, the acACS could be expressed as:

$$\varphi_i^\lambda = \frac{1}{L-\lambda}\sum_{k=1}^{L-\lambda}\left[A_k^i - A_{k+\lambda}^i\right]\left(i = {}^{15}N, {}^{13}C_\alpha, {}^{1}H_\alpha, {}^{1}H_N; \lambda < L\right) \quad (10)$$

$$P = \left[\varphi_i^0, \varphi_i^1, \varphi_i^2, \ldots, \varphi_i^\lambda\right]\left(i = {}^{15}N, {}^{13}C_\alpha, {}^{1}H_\alpha, {}^{1}H_N\right) \quad (11)$$

Synthetic Minority Over-Sampling Technique

The numbers of non-lyases are about 4.5 times that of lyases, and this leads to imbalanced data classification problems. In order to overcome this problem, we used SMOTE to solve the problem of imbalance. SMOTE is an over-sampling approach for imbalanced data classification (Wang et al., 2018a; Zhou et al., 2019). The algorithm of SMOTE is described as follows: (1) randomly choose the samples x_i from the minority class, and calculate the Euclidean distance to all other samples in this class, then K nearest neighbors of this sample were selected, (2) select x_i samples from the k nearest neighbors, and (3) generate a new sample x_{new} by: $x_{new} = x_i + \alpha\,(x - x_i)$, α is a random number in (0, 1). In this paper, the protein numbers of lyases and non-lyases are in equilibrium with SMOTE.

Feature Selection

Redundant or irrelevant features will decrease the accuracy of prediction and increase computational time. In order to remove redundant or irrelevant features, a variety of feature selection techniques have been proposed: the analysis of variance (ANOVA) (Tan et al., 2018; Li et al., 2019; Zhang et al., 2020a), Max-Relevance-Max-Distance algorithms (MRMD) (Zou et al., 2016; Wan et al., 2017; Ru et al., 2019; Kwon et al., 2020), and Minimal-Redundancy-Maximal-Relevance (MRMR) (Jiao and Du, 2016; Xu et al., 2016; Wang et al., 2018b; Kabir et al., 2020) are the representative feature selection algorithms. In this study, we selected features using the F-score algorithm; the F-score algorithm was proposed by Yi-Wei (Chen and Lin, 2006). All features are ranked according to F-score values; a higher score indicates a higher likelihood that this feature is more discriminative (Zhang et al., 2020b). It can be calculated as:

$$F_i = \frac{\left(\bar{x}_i^{(+)} - \bar{x}_i\right)^2 + \left(\bar{x}_i^{(-)} - \bar{x}_i\right)^2}{\sum_{k=1}^{n^+}\left(\bar{x}_{k,i}^{(+)} - \bar{x}_i^{(+)}\right)^2 + \frac{1}{n^- - 1}\sum_{k=1}^{n^-}\left(\bar{x}_{k,i}^{(-)} - \bar{x}_i^{(-)}\right)^2} \quad (12)$$

Where $\bar{x}_i$ is the average of the i-th feature of the whole sample, $\bar{x}_i^{(+)}$ is the average of the i-th feature of the positive samples, $\bar{x}_i^{(-)}$ is the average of the i-th feature of the negative samples; n^+ is the total number of positive samples, n^- is the total number of negative samples; $\bar{x}_{k,i}^{(+)}$ is the average of the i-th feature of the k-th sample in the positive samples, and $\bar{x}_{k,i}^{(-)}$ is the average of the i-th feature of the k-th sample in the negative samples.

To determine the optimal features, the incremental feature selection (IFS) (Ju and He, 2017; Tang et al., 2018) was employed based on the features ranked. The IFS procedure starts with one feature with the highest score, then adds features to the start feature based on their scores until all the features are added.

Support Vector Machine

The support vector machine was proposed by Vapnik; the basic idea of SVM is to transform the input data into a high-dimensional Hilbert space and then determine the optional separating hyperplane. SVM has been successfully applied in the field of computational biology and bioinformatics (Fan et al., 2013; Li and Wang, 2016; Arif et al., 2018; Chen et al., 2019; Tian et al., 2019; Wang et al., 2019a; Du et al., 2020; Jing and Li, 2020; Yang et al., 2020). Therefore, we used this classifier to build our model. The radial basis function (RBF) kernel was adopted to perform prediction. The regulation parameter c and kernel width parameter γ were tuned via the grid search method. In this paper, the LibSVM package was used to predict cell wall lytic enzymes, which can be downloaded from https://www.csie.ntu.edu.tw/~cjlin/libsvm.

[1] http://bioinf.cs.ucl.ac.uk/psipred/

[2] http://202.207.14.87:8032/bioinformation/acACS/index.asp

Performance Evaluation

In statistical prediction, three cross-validation methods are commonly used to examine a predictor for its effectiveness in practical applications: k-fold cross-validation, independent dataset test, and jackknife test (Li and Li, 2008; Tan et al., 2019; Dao et al., 2020a,b). Among the three methods, the jackknife test is deemed the most objective and rigorous. Hence, the jackknife test was used to evaluate the performance of this paper.

In order to evaluate the predictive capability and reliability of our model, the sensitivity (Sn), specificity (Sp), Matthew's correlation coefficient (MCC), and accuracy (Acc) (Bustamam et al., 2019; Cheng, 2019; Cheng et al., 2019; Feng et al., 2019; Malebary et al., 2019; Chen et al., 2020; Li and Gao, 2020; Wang et al., 2020b) were measured and defined by:

$$s_n = \frac{TP}{TP + FN} \tag{13}$$

$$s_p = \frac{TN}{TN + FN} \tag{14}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TN + FN) \times (TP + FN) \times (TN + FP)}} \tag{15}$$

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

Where TP represents the true positive, TN represents the true negative, FP represents the false positive, and FN represents the false negative.

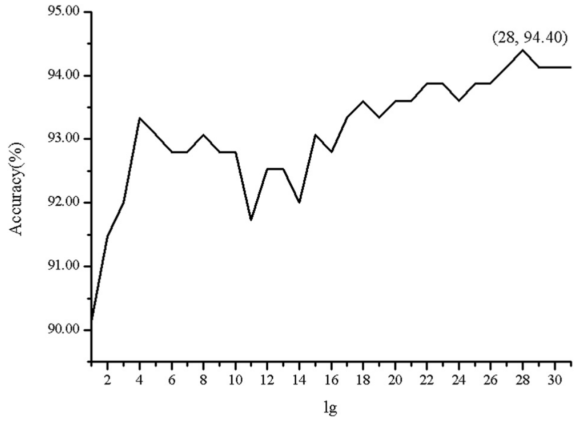


FIGURE 1 | The Acc of position-specific score matrix auto-covariance (PSSM-AC) with different *lg*.

RESULTS AND DISCUSSION

The Choice of Our Model Parameters lg, and Combination Schemes of Chemical Shifts

In order to investigate the effectiveness of the predictive model, the AAC, the DC, PSSM-AC, and the auto-covariance, average chemical shift was selected to predict the cell wall lytic enzymes. Furthermore, for the sake of the best performance of predicting

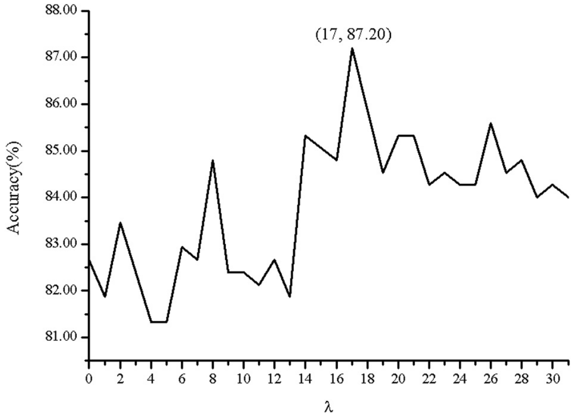


FIGURE 2 | The Acc with respect to the correlation factor λ of the combination mode of chemically shifted atoms ^{15}N, $^{13}C_{\alpha}$, $^{1}H_{\alpha}$, ^{1}H.

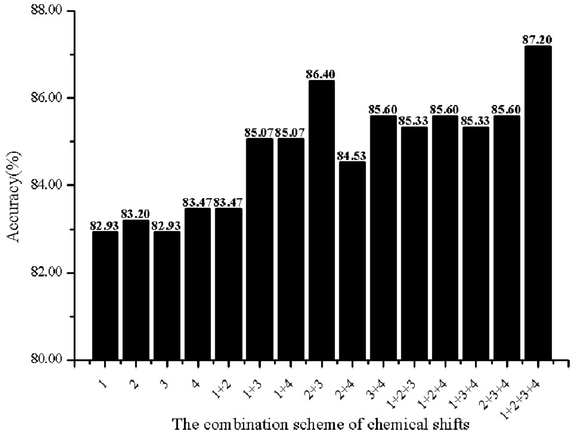


FIGURE 3 | The Acc of different combination schemes of chemical shifts. Numbers denote the chemical shifts of atoms: 1 denotes ^{15}N, 2 denotes $^{13}C_{\alpha}$, 3 denotes $^{1}H_{\alpha}$, 4 denotes $^{1}H_{N}$.

TABLE 1 | The predictive results of individual features with jackknife test by using SVM.

Features	Sn (%)	Sp (%)	MCC	Acc (%)
AAC	47.06	95.77	0.51	86.93
DC	38.24	97.39	0.48	86.67
PSSM-AC	72.06	99.67	0.81	94.40
acACS	57.35	93.81	0.55	87.20

cell wall lytic enzyme, the lg of the distance was selected, with results in **Figure 1**, and the best lg was 28 when the accuracy was the highest. In addition, the combination mode of chemically shifted atoms and the best parameter λ were selected. **Figure 2** shows that the best parameter λ was 17. The results of combination mode of chemically shifted atoms were shown in **Figure 3**; the best combination mode of chemically shifted atoms was ^{15}N, $^{13}C_{\alpha}$, $^{1}H_{\alpha}$, ^{1}H when the accuracy was the highest.

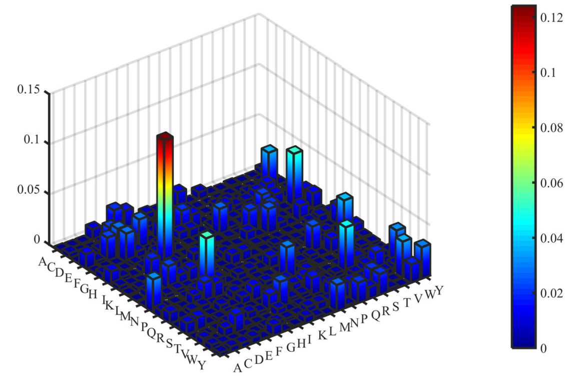


FIGURE 4 | Three-dimensional heat map of DC's F-score value.

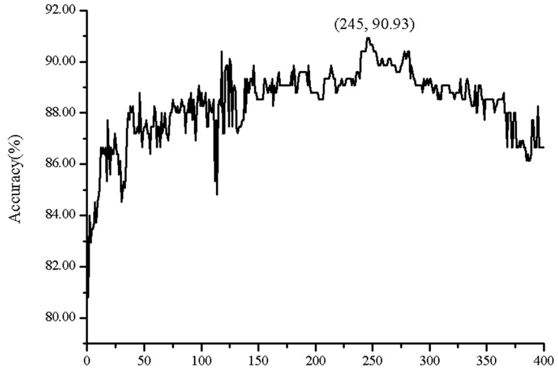


FIGURE 5 | The Acc of dipeptide composition (DC) with the incremental feature selection.

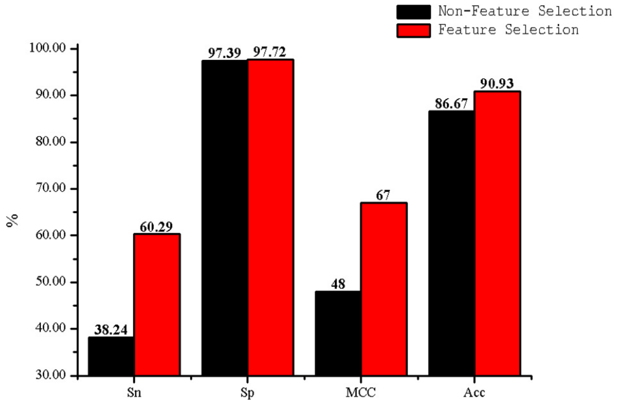


FIGURE 6 | The Acc of DC with feature selection and non-feature selection.

The Predictive Performance of Cell Wall Lytic Enzymes

The predictive performance of cell wall lytic enzymes by using the SVM classification algorithm with SMOTE was listed in **Table 1**. The highest sensitivity (Sn), specificity (Sp), Matthew's correlation coefficient (MCC), and accuracy (Acc) of individual parameters were 72.06%, 99.67%, 0.81, and 94.40% with jackknife test by using PSSM-AC. By comparison, the result of acACS was better than AAC and DC; this is probably due to the fact that

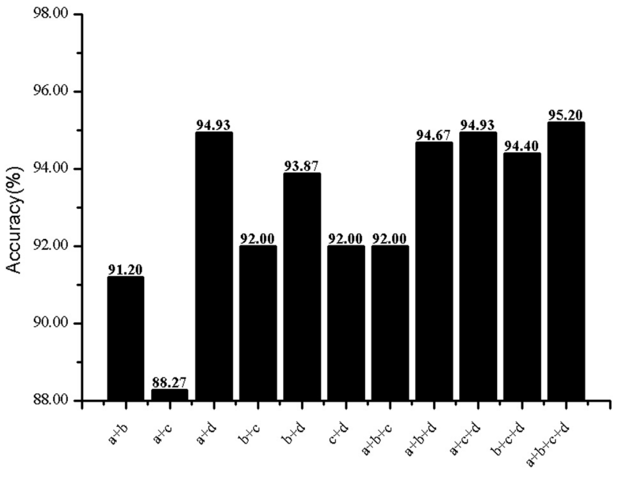


FIGURE 7 | Prediction results of different combined features. Letters denote features: a for AAC, b for DC, c for acACS, d for PSSM-AC.

TABLE 2 | The predictive results of combined feature AAC+DC+acACS+PSSM-AC by using different algorithms with and without SMOTE.

Algorithms	SMOTE (N/Y)	Sn (%)	Sp (%)	MCC	Acc (%)
SVM	N	75.00	99.67	0.83	95.20
RF		41.18	85.99	0.27	77.87
KNN		66.18	80.13	0.40	77.60
NB		86.76	66.78	0.42	70.40
SVM	Y	99.35	99.02	0.98	99.19
RF		85.99	77.52	0.64	81.76
KNN		100.00	73.94	0.77	86.97
NB		92.18	69.38	0.63	80.78

TABLE 3 | The comparison of the predictive results between this paper and existing methods.

Method	Sn (%)	Sp (%)	MCC	Acc (%)
Ding et al.	66.70	88.60	0.573	80.40
Lypred	76.47	93.16	0.678	91.30
CWLy-SVM	85.30	97.70	0.845	95.50
Our predictive model	99.35	99.02	0.98	99.19

acACS considers the protein secondary structure information. The sensitivity (Sn), Matthew's correlation coefficient (MCC), and accuracy (Acc) of AAC were all higher than DC, because DC displays redundant or irrelevant features, so we used "F-score" to select the feature. As shown in **Figure 4**, the closer the color is to red, the higher the F-score of adjacent amino acid residue and the easier it is to distinguish. On the contrary, the closer the color is to blue, the harder it is to distinguish. It can be seen that DC has some redundant information; this redundant information will reduce the prediction success rate. **Figure 5** showed the Acc of DC based on the incremental feature selection (IFS). The peak (the maximum accuracy) can be found in this curve, and it was 90.93% with 245D features. **Figure 6** showed the comparison of DC with feature selection and non-feature selection; we can see that feature selection was successfully applied to remove the irrelevant and redundant features. The Sn, MCC, and Acc were improved remarkably; Acc increased from 86.67 to 90.93%, Sn increased from 38.24 to 60.29%, and the results indicate that feature selection was helpful to enhance the predictive performance. The predictive results of different combined features with SVM without SMOTE were displayed in **Figure 7**. From **Figure 7** we can see the combined feature AAC+DC+acACS+PSSM-AC was better than other parameters. The accuracy (Acc) of combined feature AAC+DC+acACS+PSSM-AC was 95.20% with the jackknife test. This result indicates that the combined feature was powerful in the prediction of cell wall lytic enzymes.

Comparison With Different Classifiers

In order to display the power of our predictive model, our predictive model [Support Vector Machine (SVM)], Random Forest (RF), K-Nearest Neighbors (KNN), and Naive Bayes (NB) were used to predict cell wall lytic enzymes. The predictive performance of SVM, RF, KNN, and NB were listed in **Table 2**. From **Table 2**, we can see the predictive performance of SVM, RF, KNN, and NB with SMOTE were superior to those without SMOTE. The Acc of SVM, RF, KNN, and NB increased by 3.99, 3.89, 9.37, and 10.38% when using SMOTE; the MCC of SVM, RF, KNN, and NB increased by 0.15, 0.37, 0.37, and 0.21 when using SMOTE. In addition, the Sn, Sp, MCC, and Acc of SVM reached 99.35%, 99.02%, 0.98, and 99.19% by using SMOTE. The experimental results show that SVM was useful for improving the predictive performance of cell wall lytic enzymes.

Comparison With Existing Methods

To further investigate the effectiveness of our predictive model, we compared it with existing methods with the same dataset. The comparison results were listed in **Table 3**. From **Table 3**, we can see that the predictive results of cell wall lytic enzymes in our predictive model were better than those of the other methods. Furthermore, the Sn, Sp, MCC, and Acc in our predictive model reached 99.35%, 99.02%, 0.98, and 99.19%, which were 32.65%, 10.42%, 0.407, and 18.79% higher than the Ding et al. (2009) method, 22.88%, 5.86%, 0.302, and 7.89% higher than Lypred, and 14.05%, 1.32%, 0.135, and 3.69% higher than CWLy-SVM. These results indicate that our predictive model was superior to existing methods.

CONCLUSION

With the rapid rise of antibiotic-resistant strains, cell wall lytic enzymes used to destroy bacteria is a viable alternative method to avoid the crisis of antimicrobial resistance. In this work, a reliable and effective computational method was developed to identify the cell wall lytic enzymes. This model was derived from the SVM machine learning algorithm; SMOTE was used to counter the imbalanced data classification problems, and the F-score algorithm was used to remove redundant or irrelevant features. A series of experiments demonstrated that the proposed method is powerful. This method has good capability for distinguishing lyases.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: http://lin-group.cn/server/Lypred/data.html.

AUTHOR CONTRIBUTIONS

F-ML conceived the selection of feature parameters and performed the results analysis. X-YJ carried out the computation and wrote the manuscript. Both authors reviewed the manuscript.

FUNDING

This work was supported by the Natural Science Foundation of the Inner Mongolia of China (2019MS03015) and the National Natural Science Foundation of China (31360206).

REFERENCES

Ajuebor, J., McAuliffe, O., O'Mahony, J., Ross, R. P., Hill, C., and Coffey, A. (2016). Bacteriophage endolysins and their applications. *Sci. Prog.* 99, 183–199. doi: 10.3184/003685016x14627913637705

Arif, M., Hayat, M., and Jan, Z. (2018). iMem-2LSAAC: a two-level model for discrimination of membrane proteins and their types by extending the notion of SAAC into chou's pseudo amino acid composition. *J. Theor. Biol.* 442, 11–21. doi: 10.1016/j.jtbi.2018.01.008

Bhagwat, A., Collins, C. H., and Dordick, J. S. (2019). Selective antimicrobial activity of cell lytic enzymes in a bacterial consortium. *Appl. Microbiol. Biotechnol.* 103, 7041–7054. doi: 10.1007/s00253-019-09955-0

Bustamam, A., Musti, M. I. S., Hartomo, S., Aprilia, S., Tampubolon, P. P., and Lestari, D. (2019). Performance of rotation forest ensemble classifier and feature extractor in predicting protein interactions using amino acid sequences. *BMC Genomics* 20:950. doi: 10.1186/s12864-019-6304-y

Chen, J., Zhao, J., Yang, S., Chen, Z., and Zhang, Z. (2019). Prediction of protein ubiquitination sites in *Arabidopsis thaliana*. *Curr. Bioinform.* 14, 614–620. doi: 10.2174/1574893614666190311141647

Chen, W., Feng, P., and Nie, F. (2020). iATP: a sequence based method for identifying anti-tubercular peptides. *Med. Chem.* 16, 620–625. doi: 10.2174/1573406415666191002152441

Chen, X., Tang, H., Li, W., Wu, H., Chen, W., Ding, H., et al. (2016). Identification of bacterial cell wall lyases via pseudo amino acid composition. *Biomed. Res. Int.* 2016, 1654623–1654623. doi: 10.1155/2016/1654623

Chen, Y. W., and Lin, C. J. (2006). "Combining SVMs with various feature selection strategies," in *Feature Extraction. Studies in Fuzziness and Soft Computing*, Vol. 207, eds I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh (Berlin: Springer).

Cheng, L. (2019). Computational and biological methods for gene therapy. *Curr. Gene Ther.* 19:210. doi: 10.2174/156652321904191022113307

Cheng, L., Qi, C., Zhuang, H., Fu, T., and Zhang, X. (2020). gutMDisorder: a comprehensive database for dysbiosis of the gut microbiota in disorders and interventions. *Nucleic Acids Res.* 48, D554–D560. doi: 10.1093/nar/gkz843

Cheng, L., Zhao, H., Wang, P., Zhou, W., Luo, M., Li, T., et al. (2019). Computational Methods for Identifying Similar Diseases. *Mol. Ther. Nucleic Acids* 18, 590–604. doi: 10.1016/j.omtn.2019.09.019

Dao, F. Y., Lv, H., Yang, Y. H., Zulfiqar, H., Gao, H., and Lin, H. (2020a). Computational identification of N6-methyladenosine sites in multiple tissues of mammals. *Comput. Struct. Biotechnol. J.* 18, 1084–1091. doi: 10.1016/j.csbj.2020.04.015

Dao, F. Y., Lv, H., Zulfiqar, H., Yang, H., Su, W., Gao, H., et al. (2020b). A computational platform to identify origins of replication sites in eukaryotes. *Brief. Bioinform.* bbaa017. doi: 10.1093/bib/bbaa017

Der Ploeg, J. R. V. (2008). Characterization of *Streptococcus gordonii* prophage PH15: complete genome sequence and functional analysis of phage-encoded integrase and endolysin. *Microbiology* 154, 2970–2978. doi: 10.1099/mic.0.2008/018739-0

Ding, H., Luo, L., and Lin, H. (2009). Prediction of cell wall lytic enzymes using Chou's amphiphilic pseudo amino acid composition. *Protein Pept. Lett.* 16, 351–355. doi: 10.2174/092986609787848045

Dong, Q., Zhou, S., and Guan, J. (2009). A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics* 25, 2655–2662. doi: 10.1093/bioinformatics/btp500

Du, L., Meng, Q., Chen, Y., and Wu, P. (2020). Subcellular location prediction of apoptosis proteins using two novel feature extraction methods based on evolutionary information and LDA. *BMC Bioinform.* 21:212. doi: 10.1186/s12859-020-3539-1

Fan, G. L., Li, Q. Z., and Zuo, Y. C. (2013). Predicting acidic and alkaline enzymes by incorporating the average chemical shift and gene ontology informations into the general form of Chou's PseAAC. *Process Biochem.* 48, 1048–1053. doi: 10.1016/j.procbio.2013.05.012

Fan, G. L., Liu, Y. L., Zuo, Y. C., Mei, H. X., Rang, Y., Hou, B. Y., et al. (2014). acACS: improving the prediction accuracy of protein subcellular locations and protein classification by incorporating the average chemical shifts composition. *Sci. World J.* 2014:864135. doi: 10.1155/2014/864135

Feng, P., Xu, Z., Yang, H., Lv, H., Ding, H., and Liu, L. (2019). Identification of D modification sites by integrating heterogeneous features in *Saccharomyces cerevisiae*. *Molecules* 24:380. doi: 10.3390/molecules24030380

Gondil, V. S., Harjai, K., and Chhibber, S. (2020). Endolysins as emerging alternative therapeutic agents to counter drug-resistant infections. *Int. J. Antimicrob. Agents* 55:105844. doi: 10.1016/j.ijantimicag.2019.11.001

Huang, G., and Li, J. (2018). Feature extractions for computationally predicting protein post- translational modifications. *Curr. Bioinform.* 12, 387–395. doi: 10.2174/1574893612666170707094916

Jiao, Y. S., and Du, P. F. (2016). Prediction of golgi-resident protein types using general form of chou's pseudo-amino acid compositions: approaches with minimal redundancy maximal relevance feature selection. *J. Theor. Biol.* 402, 38–44. doi: 10.1016/j.jtbi.2016.04.032

Jing, X. Y., and Li, F. M. (2020). Identifying heat shock protein families from imbalanced data by using combined features. *Comput. Math. Methods Med.* 2020:8894478. doi: 10.1155/2020/8894478

Ju, Z., and He, J. J. (2017). Prediction of lysine propionylation sites using biased SVM and incorporating four different sequence features into Chou's PseAAC. *J. Mol. Graph. Model.* 76, 356–363. doi: 10.1016/j.jmgm.2017.07.022

Kabir, M., Ahmad, S., Iqbal, M., and Hayat, M. (2020). iNR-2L: a two-level sequence-based predictor developed via Chou's 5-steps rule and general PseAAC for identifying nuclear receptors and their families. *Genomics* 112, 276–285. doi: 10.1016/j.ygeno.2019.02.006

Kalali, Y., Haghighat, S., and Mahdavi, M. (2019). Passive immunotherapy with specific IgG fraction against autolysin: analogous protectivity in the MRSA infection with antibiotic therapy. *Immunol. Lett.* 212, 125–131. doi: 10.1016/j.imlet.2018.11.010

Kwon, E., Cho, M., Kim, H., and Son, H. S. (2020). A study on host tropism determinants of influenza virus using machine learning. *Curr. Bioinform.* 15, 121–134. doi: 10.2174/1574893614666191104160927

Li, F. M., and Gao, X. W. (2020). Predicting gram-positive bacterial protein subcellular location by using combined features. *Biomed. Res. Int.* 2020:9701734. doi: 10.1155/2020/9701734

Li, F. M., and Li, Q. Z. (2008). Predicting protein subcellular location using Chou's pseudo amino acid composition and improved hybrid approach. *Protein Pept. Lett.* 15, 612–616. doi: 10.2174/092986608784966930

Li, F. M., and Wang, X. Q. (2016). Identifying anticancer peptides by using improved hybrid compositions. *Sci. Rep.* 6:33910. doi: 10.1038/srep33910

Li, S., Zhang, J., Zhao, Y., Dao, F., Ding, H., Chen, W., et al. (2019). iPhoPred: a predictor for identifying phosphorylation sites in human protein. *IEEE Access* 7, 177517–177528. doi: 10.1109/ACCESS.2019.2953951

Li, W., and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* 22, 1658–1659. doi: 10.1093/bioinformatics/btl158

Malebary, S. J., Rehman, M. S. U., and Khan, Y. D. (2019). iCrotoK-PseAAC: identify lysine crotonylation sites by blending position relative statistical features according to the Chou's 5-step rule. *PloS One* 14:e0223993. doi: 10.1371/journal.pone.0223993

Meng, C., Guo, F., and Zou, Q. (2020). CWLy-SVM: a support vector machine-based tool for identifying cell wall lytic enzymes. *Comput. Biol. Chem.* 87:107304. doi: 10.1016/j.compbiolchem.2020.107304

Ru, X., Li, L., and Wang, C. (2019). Identification of phage viral proteins with hybrid sequence features. *Front. Microbiol.* 10:507. doi: 10.3389/fmicb.2019.00507

Shavrina, M. S., Zimin, A. A., Molochkov, N. V., Chernyshov, S. V., Machulin, A. V., and Mikoulinskaia, G. V. (2016). In vitro study of the antibacterial effect of the bacteriophage T5 thermostable endolysin on *Escherichia coli* cells. *J. Appl. Microbiol.* 121, 1282–1290. doi: 10.1111/jam.13251

Sibley, A. B., Cosman, M., and Krishnan, V. V. (2003). An empirical correlation between secondary structure content and averaged chemical shifts in proteins. *Biophys. J.* 84, 1223–1227. doi: 10.1016/s0006-3495(03)74937-6

Sommer, M. O. A., Munck, C., Toftkehler, R. V., and Andersson, D. I. (2017). Prediction of antibiotic resistance: time for a new preclinical paradigm? *Nat. Rev. Microbiol.* 15, 689–696. doi: 10.1038/nrmicro.2017.75

Tan, J. X., Dao, F. Y., Lv, H., Feng, P. M., and Ding, H. (2018). Identifying phage virion proteins by using two-step feature selection methods. *Molecules* 23:2000. doi: 10.3390/molecules23082000

Tan, J. X., Li, S. H., Zhang, Z. M., Chen, C. X., Chen, W., Tang, H., et al. (2019). Identification of hormone binding proteins based on machine learning methods. *Math. Biosci. Eng.* 16, 2466–2480. doi: 10.3934/mbe.2019123

Tang, H., Zhao, Y. W., Zou, P., Zhang, C. M., Chen, R., Huang, P., et al. (2018). HBPred: a tool to identify growth hormone-binding proteins. *Int. J. Biol. Sci.* 14, 957–964. doi: 10.7150/ijbs.24174

Tian, B., Wu, X., Chen, C., Qiu, W., Ma, Q., and Yu, B. (2019). Predicting protein-protein interactions by fusing various Chou's pseudo components and using wavelet denoising approach. *J. Theor. Biol.* 462, 329–346. doi: 10.1016/j.jtbi.2018.11.011

Usobiaga, P., Medrano, F. J., Gasset, M., Garcia, J. L., Saiz, J. L., Rivas, G., et al. (1996). Structural organization of the major autolysin from *Streptococcus pneumoniae*. *J. Biol. Chem.* 271, 6832–6838. doi: 10.1074/jbc.271.12.6832

Wan, S., Duan, Y., and Zou, Q. (2017). HPSLPred: an ensemble multi-label classifier for human protein subcellular location prediction with imbalanced source. *Proteomics* 17, 17–18. doi: 10.1002/pmic.201700262

Wang, C., Li, J., Liu, X., and Guo, M. (2019a). Predicting sub-Golgi apparatus resident protein with primary sequence hybrid features. *IEEE Access* 8, 4442–4450. doi: 10.1109/ACCESS.2019.2962821

Wang, J., Dai, W., Li, J., Xie, R., Dunstan, R. A., Stubenrauch, C., et al. (2020a). PaCRISPR: a server for predicting and visualizing anti-CRISPR proteins. *Nucleic Acids Res.* 48, W348–W357. doi: 10.1093/nar/gkaa432

Wang, J., Yang, B., An, Y., Marquez-Lago, T., Leier, A., Wilksch, J., et al. (2019b). Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches. *Brief. Bioinform.* 20, 931–951. doi: 10.1093/bib/bbx164

Wang, S., Wang, D., Li, J., Huang, T., and Cai, Y. D. (2018a). Identification and analysis of the cleavage site in a signal peptide using SMOTE, dagging, and feature selection methods. *Mol. Omics* 14, 64–73. doi: 10.1039/c7mo00030h

Wang, S., Zhang, Q., Lu, J., and Cai, Y. (2018b). Analysis and prediction of nitrated tyrosine sites with the mRMR method and support vector machine algorithm. *Curr. Bioinform.* 13, 3–13. doi: 10.2174/1574893611666160608075753

Wang, X. F., Gao, P., Liu, Y. F., Li, H. F., and Lu, F. (2020b). Predicting thermophilic proteins by machine learning. *Curr. Bioinform.* 15, 493–502. doi: 10.2174/1574893615666200207094357

Wu, X., Kwon, S. J., Kim, J., Kane, R. S., and Dordick, J. S. (2017). Biocatalytic Nanocomposites for Combating Bacterial Pathogens. *Annu. Rev. Chem. Biomol. Eng.* 8, 87–113. doi: 10.1146/annurev-chembioeng-060816-101612

Xu, Y., Ding, Y. X., Ding, J., Wu, L. Y., and Xue, Y. (2016). Mal-Lys: prediction of lysine malonylation sites in proteins integrated sequence-based features with mRMR feature selection. *Sci. Rep.* 6:38318. doi: 10.1038/srep38318

Yan, G., Yang, R., Fan, K., Dong, H., Gao, C., Wang, S., et al. (2019). External lysis of *Escherichia coli* by a bacteriophage endolysin modified with hydrophobic amino acids. *AMB Express* 9:106. doi: 10.1186/s13568-019-0838-x

Yang, H., Yang, W., Dao, F. Y., Lv, H., Ding, H., Chen, W., et al. (2020). A comparison and assessment of computational method for identifying recombination hotspots in *Saccharomyces cerevisiae*. *Brief. Bioinform.* 21, 1568–1580. doi: 10.1093/bib/bbz123

Yang, W., Zhu, X. J., Huang, J., Ding, H., and Lin, H. (2019). A brief survey of machine learning methods in protein sub-golgi localization. *Curr. Bioinform.* 14, 234–240. doi: 10.2174/1574893613666181113131415

Yu, J. (1997). Microbial cell wall lytic enzymes which can be used for industrial and pharmaceutical uses. *Food Sci. Biotechnol.* 6, 65–66.

Zhang, H., Xi, Q., Huang, S., Zheng, L., Yang, W., and Zuo, Y. (2020a). iSP-RAAC: identify secretory proteins of malaria parasite using reduced amino acid composition. *Comb. Chem. High Throughput Screen.* 23, 536–545. doi: 10.2174/1386207323666200402084518

Zhang, J., and Liu, B. (2019). A review on the recent developments of sequence-based protein feature extraction methods. *Curr. Bioinform.* 14, 190–199. doi: 10.2174/1574893614666181212102749

Zhang, Z., Yang, Y.-H., Ding, H., Wang, D., Chen, W., and Lin, H. (2020b). Design powerful predictor for mRNA subcellular location prediction in Homo sapiens. *Brief. Bioinform* bbz177. doi: 10.1093/bib/bbz177

Zhou, H., Chen, C., Wang, M., Ma, Q., and Yu, B. (2019). Predicting golgi-resident protein types using conditional covariance minimization with XGBoost based on multiple features fusion. *IEEE Access* 7, 144154–144164. doi: 10.1109/ACCESS.2019.2938081

Zhu, X., Feng, C., Lai, H., Chen, W., and Hao, L. (2018). Predicting protein structural classes for low-similarity sequences by evaluating different features. *Knowl. Based Syst* 163, 787–793. doi: 10.1016/j.knosys.2018.10.007

Zou, L., Nan, C., and Hu, F. (2013). Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics* 29, 3135–3142. doi: 10.1093/bioinformatics/btt554

Zou, Q., Zeng, J., Cao, L., and Ji, R. (2016). A novel features ranking metric with application to scalable visual and bioinformatics data classification. *Neurocomputing* 173, 346–354. doi: 10.1016/j.neucom.2014.12.123

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers
in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 21 January 2021
doi: 10.3389/fbioe.2020.645520

prPred: A Predictor to Identify Plant Resistance Proteins by Incorporating k-Spaced Amino Acid (Group) Pairs

Yansu Wang[1†], Pingping Wang[2†], Yingjie Guo[1], Shan Huang[3], Yu Chen[4*] and Lei Xu[1*]

[1] School of Electronic and Communication Engineering, Shenzhen Polytechnic, Shenzhen, China, [2] School of Life Science and Technology, Harbin Institute of Technology, Harbin, China, [3] Department of Neurology, The 2nd Affiliated Hospital of Harbin Medical University, Harbin, China, [4] College of Information and Computer Engineering, Northeast Forestry University, Harbin, China

To infect plants successfully, pathogens adopt various strategies to overcome their physical and chemical barriers and interfere with the plant immune system. Plants deploy a large number of resistance (R) proteins to detect invading pathogens. The R proteins are encoded by resistance genes that contain cell surface-localized receptors and intracellular receptors. In this study, a new plant R protein predictor called prPred was developed based on a support vector machine (SVM), which can accurately distinguish plant R proteins from other proteins. Experimental results showed that the accuracy, precision, sensitivity, specificity, F1-score, MCC, and AUC of prPred were 0.935, 1.000, 0.806, 1.000, 0.893, 0.857, and 0.948, respectively, on an independent test set. Moreover, the predictor integrated the HMMscan search tool and Phobius to identify protein domain families and transmembrane protein regions to differentiate subclasses of R proteins. prPred is available at https://github.com/Wangys-prog/prPred. The tool requires a valid Python installation and is run from the command line.

Keywords: prPred, plant R protein, CKSAAP, CKSAAGP, support vector machine

OPEN ACCESS

Edited by:
Zhibin Lv,
University of Electronic Science and Technology of China, China

Reviewed by:
Fei Guo,
Tianjin University, China
Chunyu Wang,
Harbin Institute of Technology, China

****Correspondence:***
Yu Chen
nefu_chenyu@163.com
Lei Xu
csleixu@szpt.edu.cn

†These authors have contributed equally to this work

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *23 December 2020*
Accepted: *31 December 2020*
Published: *21 January 2021*

Citation:
Wang Y, Wang P, Guo Y, Huang S, Chen Y and Xu L (2021) prPred: A Predictor to Identify Plant Resistance Proteins by Incorporating k-Spaced Amino Acid (Group) Pairs. Front. Bioeng. Biotechnol. 8:645520. doi: 10.3389/fbioe.2020.645520

INTRODUCTION

Plant pathogens can disturb the plant immune system to support their growth and development within plant tissue. The propagation and spread of pathogens threaten food security and cause crop and economic losses. To recognize invading pathogens, plants have evolved various disease resistance proteins (R proteins). There are two main categories of plant R proteins: membrane-bound pattern recognition receptors (PRRs) and intracellular resistance receptors. PRRs are comprised of two receptor classes, receptor-like proteins (RLPs) and receptor-like kinases (RLKs), that are located on the plant plasma membrane as the first layer of the surveillance system to detect microbe-derived molecular patterns. PRRs typically contain highly variable extracellular domains, such as lysin motif (LysM), leucine-rich repeat (LRR), and lectin domains (Zhou and Yang, 2016). The majority of intracellular resistance receptors (NBS-LRRs or NLRs) are nucleotide-binding sites (NBSs) and LRR proteins that can recognize effectors delivered into host cells by pathogens. The NBS domain is part of the NB-ARC domain that contains additional subdomains, including apoptotic protease-activating factor-1 (APAF-1), R gene products and caenorhabditis elegans death-4 protein (CED-4) (van der Biezen and Jones, 1998; Van Ooijen et al., 2008). NLR proteins are divided into two subclasses based on the N-terminal structure: TIR-NBS-LRR (TNL), which contains a toll-like-interleukin receptor (TIR) domain, and CC-NBS-LRR, which carries a coiled-coil (CC) domain (Han, 2019; Sun et al., 2020).

Five computational approaches have been developed for R protein prediction (**Table 1**). NLR-parser, RGAugury, and Restrepo-Montoya's pipeline are alignment-based tools, and NBSPred and DRPPP are learning-based tools. NLR-parser uses motif alignment and search tool (MAST) to identify NLR-like sequences (Steuernagel et al., 2015). RGAugury identifies different subclasses of R proteins, including membrane-associated receptors (RLPs or RLKs) and NBS-containing proteins, by integrating the results generated from several computing programs, such as BLAST (Camacho et al., 2009), InterProScan (Zdobnov and Apweiler, 2001), HMMER3 (Eddy, 2011), nCoil (Lupas et al., 1991), and Phobius (Käll et al., 2004). Restrepo-Montoya et al. (2020) developed a computational approach to classify RLK and RLP proteins using SignalP 4.0 (Petersen et al., 2011), TMHMM 2 (Krogh et al., 2001) and PfamScan (Finn et al., 2014). However, methods based on sequence alignment are low-sensitive and time-consuming, which can lead to difficulties in predicting low similarity proteins. Machine learning-based methods, NBSPred and DRPPP, are used for the detection of R proteins based on SVM by considering various numerical representation schemes of protein sequences. NBSPred was developed to differentiate NLR/NLR-like proteins from non-NLR proteins. However, the NBSPred training datasets were generated by electronic searches and were not experimentally verified, which might reduce the accuracy of the model. DRPPP was built by extracting various features from input protein sequences, and the model achieved 91.11% accuracy for prediction plant R proteins. Unfortunately, the NBSPred[1] and DRPPP[2] web servers are no longer available.

In this study, we developed an accurate computational approach for identifying R proteins using various sequence features. It is worth highlighting that the composition of k-spaced amino acid pairs (CKSAAPs) and k-spaced amino acid group pairs (CKSAAGPs) were also considered in the training process. The two-step feature selection strategy was adopted to detect irrelevant and redundant features. Then, the optimal *k* value and algorithm were evaluated for R protein prediction. Ultimately, support vector machine (SVM) and 5-spaced amino acid (group) pairs were chosen and applied to construct classifiers with sequence features.

MATERIALS AND METHODS

A flowchart of our method is shown in **Figure 1**. It can be summarized in five steps: (1) data collection; (2) feature construction; (3) two-step feature selection; (4) performance evaluation of features with or without CKSAAPs and CKSAAGPs; and (5) performance evaluation of different algorithms.

Data Collection

We obtained plant R protein sequences from the PRGdb database[3]. R protein sequences were derived from 35 plant species and served as a positive dataset (Osuna-Cruz et al., 2018). Next, the known protein sequences of 35 plant species were downloaded from the NCBI protein database to construct a negative dataset. The sequences containing NB-ARC, LRR, Pkinase, TIR, FNIP, Acalin, peptidase_C48, PPR, zf-BED, and WRKY were filtered by a Pfam domain search (Kushwaha et al., 2016). To remove redundancy, proteins with sequence similarity >30% were excluded from the non-R protein dataset using CD-HIT (Fu et al., 2012). However, 34,975 protein sequences remained in the non-R protein dataset after filtering, thus, to ensure the balance of data, 304 protein sequences were selected randomly from the identified non-R proteins to serve as a final negative dataset. Then, 152 R proteins and 304 non-R proteins were split into training and test datasets at an 8:2 ratio. Finally, the training dataset is made up of 121 R protein sequences and 243 non-R protein sequences, and the independent test dataset is composed of 31 R protein sequences and 61 non-R protein sequences.

Feature Construction

Features were extracted from input sequences using iFeature (Chen et al., 2018), such as amino acid composition, grouped amino acid composition, quasi-sequence-order, composition/transition/distribution (C/T/D), autocorrelation, conjoint triad and pseudo-amino acid composition (PseAAC). More detailed information about the features is described in the **Supplementary Methods** and **Supplementary Table 1**.

There are lots of feature extraction methods (Pal et al., 2016; Zeng et al., 2016; Liao et al., 2018; Zhang and Liu, 2019; Ikram et al., 2020; Li J. et al., 2020; Wang et al., 2020; Zhao et al., 2020; Zhu et al., 2020). In this work, we utilized CKSAAPs and CKSAAGPs as numeric vectors to represent the protein sequence. CKSAAP was used to calculate the occurrence frequencies of any two amino acids separated by any k amino acids. For example, if *k* = 0, the 0-spaced residue pairs can be represented as: AA, AC, AD, . . ., YY; if *k* = 1, the 1-spaced residue pairs can be expressed as AxA, AxC, AxD, . . ., YxY. The CKSAAPs are defined as:

$$k = 0 \left(\frac{\mathrm{N}\,[\mathrm{AA}]}{N_0}, \frac{\mathrm{N}\,[\mathrm{AC}]}{N_0}, \frac{\mathrm{N}\,[\mathrm{AD}]}{N_0}, \ \ldots\ldots, \ \frac{\mathrm{N}\,[\mathrm{YY}]}{N_0} \right) 400$$

$$k = 1 \left(\frac{\mathrm{N}\,[\mathrm{AxA}]}{N_1}, \frac{\mathrm{N}\,[\mathrm{AxC}]}{N_1}, \frac{\mathrm{N}\,[\mathrm{AxD}]}{N_1}, \ \ldots\ldots, \ \frac{\mathrm{N}\,[\mathrm{YxY}]}{N_1} \right) 400$$

[1] http://soilecology.biol.lu.se/nbs/

[2] http://14.139.240.55/NGS/download.php

[3] http://prgdb.org/prgdb/

TABLE 1 | Summary of existing tools for plant R protein prediction.

Tool	Methods	Objects	Sites	References
NLR-parser	Motif alignment and search tool (MAST)	NLRs	http://github.com/steuernb/NLR-Parser	Steuernagel et al., 2015
RGAugury	BLAST search and domain/motif analysis	RLKs, RLPs, NLRs	https://bitbucket.org/yaanlpc/rgaugury	Li et al., 2016
Restrepo-Montoya's method	BLAST search and domain/motif analysis	RLKs, RLPs	https://github.com/drestmont/plant_rlk_rlp/	Restrepo-Montoya et al., 2020
NBSPred	SVM	NLRs	http://soilecology.biol.lu.se/nbs/	Kushwaha et al., 2016
DRPPP	SVM	R proteins	http://14.139.240.55/NGS/download.php	Pal et al., 2016

SVM, support vector machine.

$$k=2\left(\frac{\mathrm{N}\,[\mathrm{AxxA}]}{N_2},\ \frac{\mathrm{N}\,[\mathrm{AxxC}]}{N_2},\frac{\mathrm{N}\,[\mathrm{AxxD}]}{N_2},\ldots,\frac{\mathrm{N}\,[\mathrm{YxxY}]}{N_2}\right)400$$

where "x" represents any of 20 amino acids; N_k was calculated as $N_k = L - (k+1)$, $k = 1, 2, 3\ldots$, where L represents the length of a given protein sequence. The final feature vector was computed by concatenating the individual feature vectors; for example, if $k = 5$, the number of vector dimensions would be 400 × 6 = 2,400.

Amino acid residues can be divided into five categories based on chemical properties of the side chains, including aliphatic group (g1: GAVLMI), aromatic group (g2: FYW), positive charged group (g3: KRH), negative charged group (g4: DE), and uncharged group (g5: STCPNQ). k-spaced amino acid group pairs (CKSAAGP) is based on the frequency of two group separated by any k amino acids. If $k = 0$, the 0-spaced group pairs is represented as:

$$k=0\left(\frac{\mathrm{N}\left[\mathrm{g1g1}\right]}{N_0},\ \frac{\mathrm{N}\left[\mathrm{g1g2}\right]}{N_0},\frac{\mathrm{N}\left[\mathrm{g1g3}\right]}{N_0},\ \ldots\ldots,\ \frac{\mathrm{N}\left[\mathrm{g5g5}\right]}{N_0}\right)25$$

Two-Step Feature Selection Strategy

First, feature vectors were sorted according to the value of information gain (IG). A new feature list was generated in descending order of the IG value. Second, we selected or removed features based on the accuracy value during the training process. We added features from higher IG value to lower IG value. If the addition of a feature did not reduce the accuracy in the cross-validation strategy, then the feature vector was retained; otherwise, it was removed.

Machine Learning Algorithms

Eight algorithms, including logistic regression (LR) (Hosmer et al., 2013), K-nearest neighbors (KNN) (Kramer, 2013), SVM (Hearst et al., 1998), decision tree (DT) (Swain and Hauska, 1977), random forest (RF) (Breiman, 2001), gradient boosting classifier (GBC) (Aler et al., 2017), Adaboost (Schapire, 2013), and extra-tree classifier (ETC) (Geurts et al., 2006), were chosen to train the model. We applied grid search (GS) to find optimal parameter combination in 10-fold cross-validation for each model. GS requires specifying a range for parameters, for example, the SVM parameter optimization using GS is implemented within the given ranges of $C = \{-5,\ 11\}$ and $\gamma = \{-9, 13\}$.

Performance Evaluation

To estimate the contributions of CKSAAPs and CKSAAGPs and to measure the overall predictive performance of the classification models, six parameters were applied for 10-fold cross-validation and independent tests (Hearst et al., 1998; An et al., 2019; Chen et al., 2019; Ding et al., 2019a,b; Fang et al., 2019; Jiang et al., 2019; Lv et al., 2019b, 2020b; Shen et al., 2019; Liu et al., 2020), including precision (Pre), sensitivity (Sen), specificity (Spe), accuracy (Acc), F1-score, and Matthew's correlation coefficient (MCC). They are defined as follows:

$$\mathrm{Pre}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{1}$$

$$\mathrm{Sen}\ =\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{2}$$

$$\mathrm{Spe}=\frac{\mathrm{TN}}{\mathrm{FP}+\mathrm{TN}} \tag{3}$$

$$\mathrm{Acc}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}} \tag{4}$$

$$\mathrm{F1-score}=\frac{2\times\mathrm{Pre}\ \times\mathrm{Sen}}{\mathrm{Pre}+\mathrm{Sen}} \tag{5}$$

$$\mathrm{MCC}=\frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \tag{6}$$

where TP is the number of R proteins classified as R proteins, TN is the number of non-R proteins classified as non-R proteins, FP is the number of non-R proteins classified as R proteins, and FN is the number of R proteins classified as non-R proteins.

Additionally, the ROC curve and PR curve were used as visual assessment metrics. The ROC curve shows the false-positive rate versus the true positive rate, and the PR curve is recall versus precision. The area under the curve (AUC) is also provided as performance measure (Wang et al., 2010; Cheng et al., 2019). An AUC close to 1 indicates better prediction of the model.

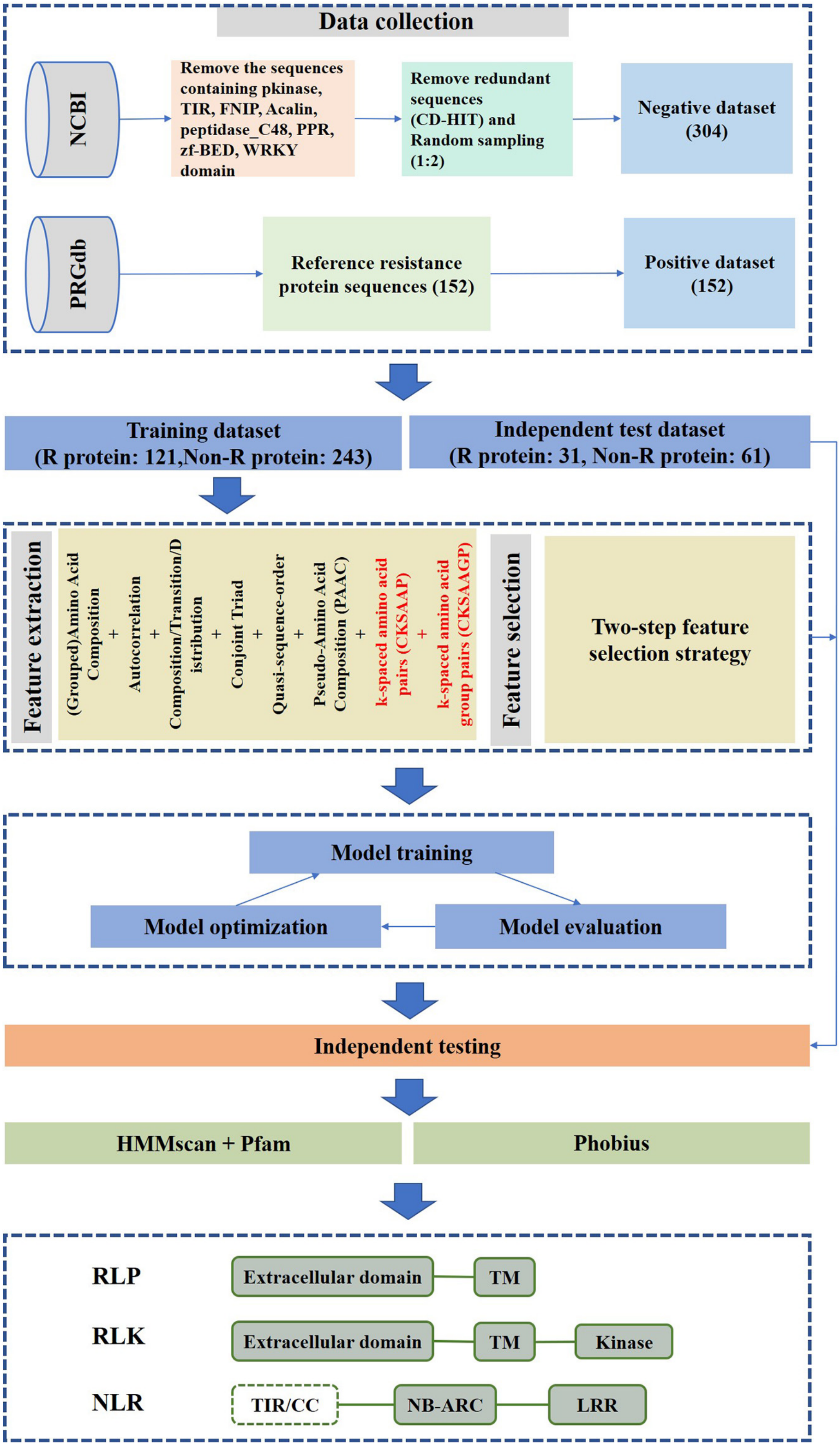

Data collection
NCBI
Remove the sequences containing pkinase, TIR, FNIP, Acalin, peptidase_C48, PPR, zf-BED, WRKY domain
Remove redundant sequences (CD-HIT) and Random sampling (1:2)
Negative dataset (304)
PRGdb
Reference resistance protein sequences (152)
Positive dataset (152)
Training dataset (R protein: 121,Non-R protein: 243)
Independent test dataset (R protein: 31, Non-R protein: 61)
Feature extraction
(Grouped)Amino Acid Composition
+
Autocorrelation
+
Composition/Transition/Distribution
+
Conjoint Triad
+
Quasi-sequence-order
+
Pseudo-Amino Acid Composition (PAAC)
+
k-spaced amino acid pairs (CKSAAP)
+
k-spaced amino acid group pairs (CKSAAGP)
Feature selection
Two-step feature selection strategy
Model training
Model optimization
Model evaluation
Independent testing
HMMscan + Pfam
Phobius
RLP
Extracellular domain
TM
RLK
Extracellular domain
TM
Kinase
NLR
TIR/CC
NB-ARC
LRR

FIGURE 1 | prPred workflow.

TABLE 2 | Performance comparison of features with and without CKSAAP and CKSAAGP in the independent dataset test.

	Algorithms	**Independent dataset test**						
		Acc	**Pre**	**Sen**	**Spe**	**F1-score**	**MCC**	**AUC**
Without CKSAAPs and CKSAAGPs	LR	0.891	0.839	0.839	0.918	0.839	0.757	0.919
	KNN	0.891	0.862	0.806	0.934	0.833	0.754	0.928
	SVM	0.902	0.893	0.806	0.951	0.847	0.778	0.935
	RF	0.880	0.885	0.742	0.951	0.807	0.727	0.924
	DT	0.859	0.846	0.710	0.934	0.772	0.676	0.847
	GBC	0.815	0.733	0.710	0.869	0.721	0.583	0.839
	Adaboost	0.848	0.840	0.677	0.934	0.750	0.650	0.859
	ETC	0.913	0.926	0.806	0.967	0.862	0.803	0.947
k = 5	LR	0.891	0.862	0.806	0.934	0.833	0.754	0.946
	KNN	0.924	0.929	0.839	0.967	0.881	0.828	0.935
	SVM	**0.935**	**1.000**	**0.806**	**1.000**	**0.893**	**0.857**	**0.948**
	RF	0.913	0.960	0.774	0.984	0.857	0.805	0.931
	DT	0.880	0.917	0.710	0.967	0.800	0.729	0.854
	GBC	0.902	0.923	0.774	0.967	0.842	0.778	0.882
	Adaboost	0.870	0.828	0.774	0.918	0.800	0.704	0.880
	ETC	0.924	0.962	0.806	0.984	0.877	0.829	0.938

LR, logistic regression; KNN, K nearest neighbors; SVM, support vector machine; RF, random forest; DT, decision tree; GBC, gradient boosting classifier; ETC, extra tree classifier. The bold values represent the predictive performance of SVM based on 5-spaced amino acid pairs.

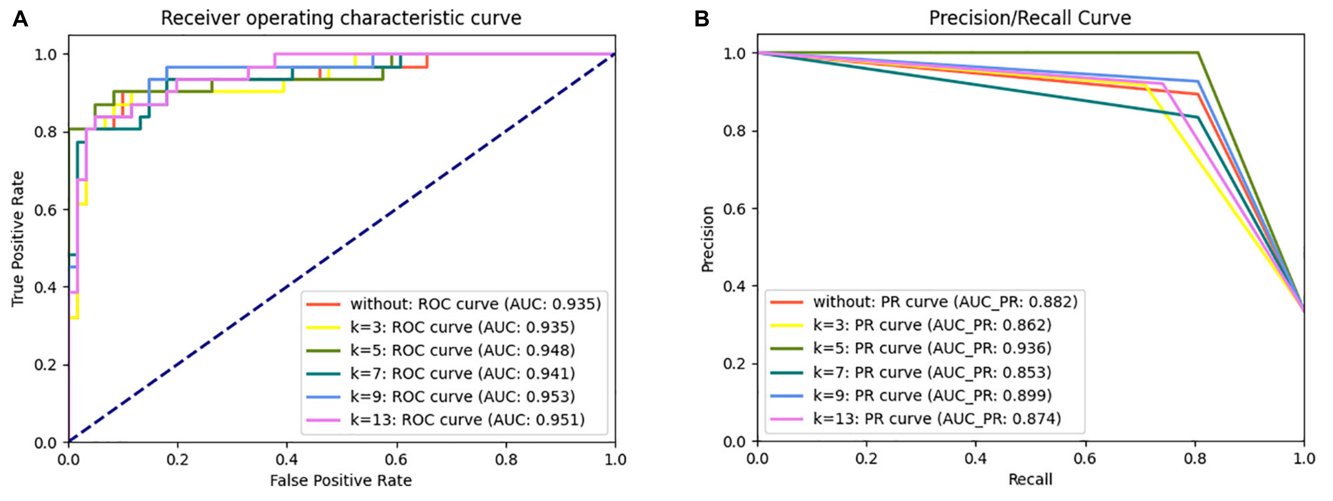


FIGURE 2 | ROC **(A)** and PR **(B)** curve for the prPred classifier in the independent dataset test.

TABLE 3 | Example results in the CSV-format output file of prPred.

ID	**R_protein_possibility**	**TM**	**SP**	**Domain**				
Protein1	0.992151981	0	0	NB-ARC (PF00931.22)	Rx_N (PF18052.1)	LRR_8 (PF13855.6)	LRR_8 (PF13855.6)	LRR_8 (PF13855.6)
Protein2	0.992149469	0	0	NB-ARC (PF00931.22)	NB-ARC (PF00931.22)	Rx_N (PF18052.1)	Rx_N (PF18052.1)	Rx_N (PF18052.1)
Protein3	0.998599022	0	0	TIR (PF01582.20)	NB-ARC (PF00931.22)	NB-ARC (PF00931.22)	TIR_2 (PF13676.6)	
Protein4	0.992166647	1	Y	Pkinase (PF00069.25)	Pkinase_Tyr (PF07714.17)	LRRNT_2 (PF08263.12)	LRRNT_2 (PF08263.12)	LRR_8 (PF13855.6)
Protein5	0.992152188	1	Y	LRR_8 (PF13855.6)	LRR_8 (PF13855.6)	LRR_8 (PF13855.6)	LRR_8 (PF13855.6)	LRR_8 (PF13855.6)
Protein6	0.023914191	0	0					
Protein7	0.022744187	0	0	FHA (PF00498.26)				
Protein8	0.023851809	1	0					

RESULTS

Comparison of Different Feature Combinations and Classification Models

CKSAAPs and CKSAAGPs are numerical encoding schemes that can capture short linear motif information, and the composition of CKSAAPs has been successfully applied to identify protein modification sites (Cheng et al., 2018; Lv et al., 2020c,d). We constructed feature vectors with CKSAAPs and CKSAAGPs because plant R proteins contain motif information distinct from that of non-R proteins (**Supplementary Figure 1**). The numerical encoding schemes of CKSAAP and CKSAAGP have exhibited obvious differences between R and non-R proteins using Wilcoxon rank sum test (**Supplementary Figure 2**). **Table 2** showed that different models had different responses to the features with or without CKSAAPs and CKSAAGPs. For example, the Acc of LR showed no noticeable changes when CKSAAP and CKSAAGP features were added, while the Acc of SVM was improved from 0.902 to 0.935 in the independent dataset when considering 5-spaced amino acid pairs.

To determine the optimal algorithms and k value, we explored the discrimination power of $k =$ 3-, 5-, 7-, 9-, and 13-spaced amino acid pairs using different algorithms (e.g., LR, KNN, SVM, RF, DT, GBC, Adaboost, and ETC) (**Supplementary Table 2**). We observed that SVM achieved better performance than other algorithms in 10-fold cross-validation tests in the same k-value. Although the AUC of SVM when $k = 5$ ($\mathrm{AUC}_{k\,=\,5} = 0.948$) was slightly lower than that when $k = 9$ and 13 ($\mathrm{AUC}_{k\,=\,9} = 0.953$, $\mathrm{AUC}_{k\,=\,13} = 0.951$) in the ROC curve in the independent dataset tests, the PR curve showed 4.12 and 7.09% improvements in AUC-PR when k = 5 compared with $k = 9$ and 13 (**Figure 2**). Moreover, the Acc, Spe, F1-score, and MCC values were improved by 2.41% (4.94%), 3.41% (3.41%), 3.60% (8.77%), and 6.72% (13.81%), respectively, compared with $k = 9$ (and 13) (**Supplementary Table 2**). Therefore, we chose SVM as the model and $k = 5$ to build the plant R protein predictor. The predictor showed satisfactory prediction results for the independent dataset with an Acc of 0.935, Pre of 1.000, Sen of 0.806, Spe of 1.000, F1-score of 0.893, MCC of 0.857, and AUC of 0.948 (**Table 2** and **Supplementary Table 2**). The optimal parameters of SVM with the RBF kernel were $C = 2.0$ and $\gamma = 0.0078$.

Prediction Pipeline of prPred

Because the published methods based on machine learning algorithms (e.g., NBSPred and DRPPP) are no longer available, performance comparisons cannot be carried out between prPred and the state-of-the-art methods. The alignment-based tools, NLR-parser and Restrepo-Montoya's method are mainly applied to predict NLRs and PRRs (RLKs and RLPs), respectively. The RGAugury project aims to identify resistance gene analogs for plant genomes using interolog- and domain-based approaches. In the study, prPred integrated machine learning method and sequence alignment-based method to analyze and evaluate the potential R proteins. Except for predicting the potential R proteins, it was capable of annotating protein domain families based on Pfam-A using a hidden Markov model (HMM) and searching transmembrane regions (TMs) using Phobius to differentiate RLPs/PLKs from NLRs. Users can import protein sequences in FASTA format, and the prPred prediction results can be saved to CSV- and FASTA-formatted file. The CSV-formatted file output contains information about the protein sequence ID, prediction probability score, TM number, that as shown in **Table 3**.

CONCLUSION

In this study, we developed a bioinformatics tool called prPred for the prediction of plant resistance proteins that combines CKSAAP and CKSAAGP features based on SVM. The predictive and analytical results demonstrated that the constructed model is an efficient predictor to distinguish R proteins from non-R proteins. CKSAAP and CKSAAGP features provide important improvements in the prediction performance. We expect that prPred will be a useful tool to facilitate biological research and provide guidance for related experimental validation. In the feature, we will use deep learning method and deep representation learning features for prPred (Lv et al., 2019a, 2020a, 2021; Li F. et al., 2020).

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/**Supplementary Material.**

AUTHOR CONTRIBUTIONS

YW and PW were responsible for experiments and manuscript preparation. YG and SH participated in discussions. YC and LX worked as supervisor for all procedures. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the Post-doctoral Foundation Project of Shenzhen Polytechnic (6020330004K).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fbioe.2020.645520/full#supplementary-material

REFERENCES

Aler, R., Galván, I. M., Ruiz-Arias, J. A., and Gueymard, C. A. (2017). Improving the separation of direct and diffuse solar radiation components using machine learning by gradient boosting. *Sol. Energy* 150, 558–569. doi: 10.1016/j.solener.2017.05.018

An, J.-Y., Zhou, Y., Zhang, L., Qiang, N., and Wang, D.-F. (2019). Improving self-interacting proteins prediction accuracy using protein evolutionary information and weighed-extreme learning machine. *Curr. Bioinform.* 14, 115–122. doi: 10.2174/1574893613666180209161152

Breiman, L. (2001). Random forests. *Mach. Learn.* 45, 5–32. doi: 10.1023/A:1010933404324

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., et al. (2009). BLAST+: architecture and applications. *BMC Bioinformatics* 10:421. doi: 10.1186/1471-2105-10-421

Chen, J., Zhao, J., Yang, S., Chen, Z., and Zhang, Z. (2019). Prediction of protein ubiquitination sites in arabidopsis thaliana. *Curr. Bioinform.* 14, 614–620. doi: 10.2174/1574893614666190311141647

Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T. T., Wang, Y., et al. (2018). iFeature: a python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics* 34, 2499–2502. doi: 10.1093/bioinformatics/bty140

Cheng, L., Hu, Y., Sun, J., Zhou, M., and Jiang, Q. (2018). DincRNA: a comprehensive web-based bioinformatics toolkit for exploring disease associations and ncRNA function. *Bioinformatics* 34, 1953–1956. doi: 10.1093/bioinformatics/bty002

Cheng, L., Wang, P., Tian, R., Wang, S., Guo, Q., Luo, M., et al. (2019). LncRNA2Target v2.0: a comprehensive database for target genes of lncRNAs in human and mouse. *Nucleic Acids Res.* 47, D140–D144.

Ding, Y., Tang, J., and Guo, F. (2019a). Identification of drug-side effect association via multiple information integration with centered kernel alignment. *Neurocomputing* 325, 211–224. doi: 10.1016/j.neucom.2018.10.028

Ding, Y., Tang, J., and Guo, F. (2019b). Identification of drug-target interactions via fuzzy bipartite local model. *Neural Comput. Appl.* 32, 10303–10319. doi: 10.1007/s00521-019-04569-z

Eddy, S. R. (2011). Accelerated profile HMM searches. *PLoS Comput. Biol.* 7:e1002195. doi: 10.1371/journal.pcbi.1002195

Fang, M., Lei, X., and Guo, L. (2019). A survey on computational methods for essential proteins and genes prediction. *Curr. Bioinform.* 14, 211–225. doi: 10.2174/1574893613666181112150422

Finn, R. D., Bateman, A., Clements, J., Coggill, P., Eberhardt, R. Y., Eddy, S. R., et al. (2014). Pfam: the protein families database. *Nucleic Acids Res.* 42, D222–D230. doi: 10.1093/nar/gkt1223

Fu, L., Niu, B., Zhu, Z., Wu, S., and Li, W. (2012). CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* 28, 3150–3152. doi: 10.1093/bioinformatics/bts565

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Mach. Learn.* 63, 3–42. doi: 10.1007/s10994-10006-16226-10991

Han, G. Z. (2019). Origin and evolution of the plant immune system. *New Phytol.* 222, 70–83. doi: 10.1111/nph.15596

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. (1998). Support vector machines. *IEEE Intell. Syst. Their Appl.* 13, 18–28. doi: 10.1109/5254.708428

Hosmer, D. W. Jr., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied Logistic Regression*. Hoboken, NJ: John Wiley & Sons.

Ikram, N., Qadir, M. A., and Afzal, M. T. (2020). SimExact - an efficient method to compute function similarity between proteins using gene ontology. *Curr. Bioinform.* 15, 318–327. doi: 10.2174/1574893614666191017092842

Jiang, M., Pei, Z., Fan, X., Jiang, J., Wang, Q., and Zhang, Z. (2019). Function analysis of human protein interactions based on a novel minimal loop algorithm. *Curr. Bioinform.* 14, 164–173. doi: 10.2174/1574893613666180906103946

Käll, L., Krogh, A., and Sonnhammer, E. L. (2004). A combined transmembrane topology and signal peptide prediction method. *J. Mol. Biol.* 338, 1027–1036. doi: 10.1016/j.jmb.2004.03.016

Kramer, O. (2013). "K-nearest neighbors," in *Dimensionality Reduction with Unsupervised Nearest Neighbors*, eds J. Kacprzyk, Warsaw, P. L. C. Jain, (Adelaide: Springer), 13–23. doi: 10.1007/1978-1003-1642-38652-38657_38652

Krogh, A., Larsson, B., Von Heijne, G., and Sonnhammer, E. L. (2001). Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J. Mol. Boil.* 305, 567–580. doi: 10.1006/jmbi.2000.4315

Kushwaha, S. K., Chauhan, P., Hedlund, K., and Ahrén, D. (2016). NBSPred: a support vector machine-based high-throughput pipeline for plant resistance protein NBSLRR prediction. *Bioinformatics* 32, 1223–1225. doi: 10.1093/bioinformatics/btv714

Li, F., Luo, M., Zhou, W., Li, J., Jin, X., Xu, Z., et al. (2020). Single cell RNA and immune repertoire profiling of COVID-19 patients reveal novel neutralizing antibody. *Protein Cell* 25, 1–5.

Li, J., Pu, Y., Tang, J., Zou, Q., and Guo, F. (2020). DeepAVP: a dual-channel deep neural network for identifying variable-length antiviral peptides. *IEEE J. Biomed. Health Inform.* 24, 3012–3019. doi: 10.1109/jbhi.2020.2977091

Li, P., Quan, X., Jia, G., Xiao, J., Cloutier, S., and You, F. M. (2016). RGAugury: a pipeline for genome-wide prediction of resistance gene analogs (RGAs) in plants. *BMC Genomics* 17:852. doi: 10.1186/s12864-12016-13197-x

Liao, Z., Wan, S., He, Y., and Quan, Z. (2018). Classification of small GTPases with hybrid protein features and advanced machine learning techniques. *Curr. Bioinform.* 13, 492–500. doi: 10.2174/1574893612666171121162552

Liu, B., Li, C., and Yan, K. (2020). DeepSVM-fold: protein fold recognition by combining support vector machines and pairwise sequence similarity scores generated by deep learning networks. *Brief. Bioinform.* 21, 1733–1741. doi: 10.1093/bib/bbz098

Lupas, A., Van Dyke, M., and Stock, J. (1991). Predicting coiled coils from protein sequences. *Science* 256, 1162–1164. doi: 10.1126/science.1252.5009.1162

Lv, H., Dao, F.-Y., Guan, Z.-X., Yang, H., Li, Y.-W., and Lin, H. (2020a). Deep-Kcr: accurate detection of lysine crotonylation sites using deep learning method. *Brief. Bioinform.* doi: 10.1093/bib/bbaa255 Online ahead of print

Lv, H., Dao, F. Y., Zhang, D., Guan, Z. X., Yang, H., Su, W., et al. (2020b). iDNA-MS: an integrated computational tool for detecting DNA modification sites in multiple genomes. *iScience* 23:100991. doi: 10.1016/j.isci.2020.100991

Lv, Z., Wang, D., Ding, H., Zhong, B., and Xu, L. (2020c). *Escherichia Coli* DNA N-4-methycytosine site prediction accuracy improved by light gradient boosting machine feature selection technology. *IEEE Access* 8, 14851–14859. doi: 10.1109/access.2020.2966576

Lv, Z., Zhang, J., Ding, H., and Zou, Q. (2020d). RF-PseU: a random forest predictor for RNA pseudouridine sites. *Front. Bioeng. Biotech.* 8:134. doi: 10.3389/fbioe.2020.00134

Lv, Z., Ao, C., and Zou, Q. (2019a). Protein function prediction: from traditional classifier to deep learning. *Proteomics* 19:1900119. doi: 10.1002/pmic.201900119

Lv, Z., Jin, S., Ding, H., and Zou, Q. (2019b). A random forest sub-golgi protein classifier optimized via dipeptide and amino acid composition features. *Front. Bioeng. Biotech.* 7:215. doi: 10.3389/fbioe.2019.00215

Lv, Z., Wang, P., Zou, Q., and Jiang, Q. (2021). Identification of sub-golgi protein localization by use of deep representation learning features. *Bioinformatics* doi: 10.1093/bioinformatics/btaa1074 Online ahead of print

Osuna-Cruz, C. M., Paytuvi-Gallart, A., Di Donato, A., Sundesha, V., Andolfo, G., Aiese Cigliano, R., et al. (2018). PRGdb 3.0: a comprehensive platform for prediction and analysis of plant disease resistance genes. *Nucleic Acids Res.* 46, D1197–D1201. doi: 10.1093/nar/gkx1119

Pal, T., Jaiswal, V., and Chauhan, R. S. (2016). DRPPP: a machine learning based tool for prediction of disease resistance proteins in plants. *Comput. Biol. Med.* 78, 42–48. doi: 10.1016/j.compbiomed.2016.09.008

Petersen, T. N., Brunak, S., Von Heijne, G., and Nielsen, H. (2011). SignalP 4.0: discriminating signal peptides from transmembrane regions. *Nat. Methods* 8:785. doi: 10.1038/nmeth.1701

Restrepo-Montoya, D., Brueggeman, R., Mcclean, P. E., and Osorno, J. M. (2020). Computational identification of receptor-like kinases "RLK" and receptor-like proteins "RLP" in legumes. *BMC Genomics* 21:459. doi: 10.1186/s12864-12020-06844-z

Schapire, R. E. (2013). "Explaining AdaBoost," in *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, eds B. Schölkopf, Z. Luo, and V. Vovk (Berlin: Springer), 37–52. doi: 10.1007/1978-1003-1642-41136-41136_41135

Shen, Y., Tang, J., and Guo, F. (2019). Identification of protein subcellular localization via integrating evolutionary and physicochemical information into Chou's general PseAAC. *J. Theor. Biol.* 462, 230–239. doi: 10.1016/j.jtbi.2018.11.012

Steuernagel, B., Jupe, F., Witek, K., Jones, J. D., and Wulff, B. B. (2015). NLR-parser: rapid annotation of plant NLR complements. *Bioinformatics* 31, 1665–1667. doi: 10.1093/bioinformatics/btv1005

Sun, Y., Zhu, Y.-X., Balint-Kurti, P. J., and Wang, G. F. (2020). Fine-tuning immunity: players and regulators for plant NLRs. *Trends Plant Sci.* 25, 695–713. doi: 10.1016/j.tplants.2020.1002.1008

Swain, P. H., and Hauska, H. (1977). The decision tree classifier: design and potential. *IEEE T. Geosci. Elect.* 15, 142–147. doi: 10.1109/TGE.1977.6498972

van der Biezen, E. A., and Jones, J. D. (1998). The NB-ARC domain: a novel signalling motif shared by plant resistance gene products and regulators of cell death in animals. *Curr. Biol.* 8, R226–R228.

Van Ooijen, G., Mayr, G., Kasiem, M. M., Albrecht, M., Cornelissen, B. J., and Takken, F. L. (2008). Structure–function analysis of the NB-ARC domain of plant disease resistance proteins. *J. Exp. Bot.* 59, 1383–1397. doi: 10.1093/jxb/ern045

Wang, G., Wang, Y., Teng, M., Zhang, D., Li, L., and Liu, Y. (2010). Signal transducers and activators of transcription-1 (STAT1) regulates microRNA transcription in interferon gamma-stimulated HeLa cells. *PLoS One* 5:e11794. doi: 10.1371/journal.pone.0011794

Wang, Z., He, W., Tang, J., and Guo, F. (2020). Identification of highest-affinity binding sites of yeast transcription factor families. *J. Chem. Inform. model.* 60, 1876–1883. doi: 10.1021/acs.jcim.9b01012

Zdobnov, E. M., and Apweiler, R. (2001). InterProScan–an integration platform for the signature-recognition methods in InterPro. *Bioinformatics* 17, 847–848. doi: 10.1093/bioinformatics/17.9.847

Zeng, J., Li, D., Wu, Y., Zou, Q., and Liu, X. (2016). An empirical study of features fusion techniques for protein-protein interaction prediction. *Curr. Bioinform.* 11, 4–12. doi: 10.2174/1574893611666151119221435

Zhang, J., and Liu, B. (2019). A review on the recent developments of sequence-based protein feature extraction methods. *Current Bioinformatics* 14, 190–199. doi: 10.2174/1574893614666181212102749

Zhao, X., Jiao, Q., Li, H., Wu, Y., Wang, H., Huang, S., et al. (2020). ECFS-DEA: an ensemble classifier-based feature selection for differential expression analysis on expression profiles. *BMC Bioinformatics* 21:43. doi: 10.1186/s12859-020-3388-y

Zhou, J.-M., and Yang, W.-C. (2016). Receptor-like kinases take center stage in plant biology. *Sci. China Life Sci.* 59:863. doi: 10.1007/s11427-016-5112-8

Zhu, H., Du, X., and Yao, Y. (2020). ConvsPPIS: identifying protein-protein interaction sites by an ensemble convolutional neural network with feature graph. *Curr. Bioinform.* 15, 368–378. doi: 10.2174/1574893614666191105155713

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers
in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 25 January 2021
doi: 10.3389/fbioe.2020.629937

TMPSS: A Deep Learning-Based Predictor for Secondary Structure and Topology Structure Prediction of Alpha-Helical Transmembrane Proteins

Zhe Liu[1,2], Yingli Gong[3], Yihang Bao[4], Yuanzhao Guo[4], Han Wang[4] and Guan Ning Lin[1,2]**

[1] Shanghai Mental Health Center, Shanghai Jiao Tong University School of Medicine, School of Biomedical Engineering, Shanghai Jiao Tong University, Shanghai, China, [2] Shanghai Key Laboratory of Psychotic Disorders, Shanghai, China, [3] College of Intelligence and Computing, Tianjin University, Tianjin, China, [4] School of Information Science and Technology, Institute of Computational Biology, Northeast Normal University, Changchun, China

Alpha transmembrane proteins (αTMPs) profoundly affect many critical biological processes and are major drug targets due to their pivotal protein functions. At present, even though the non-transmembrane secondary structures are highly relevant to the biological functions of αTMPs along with their transmembrane structures, they have not been unified to be studied yet. In this study, we present a novel computational method, TMPSS, to predict the secondary structures in non-transmembrane parts and the topology structures in transmembrane parts of αTMPs. TMPSS applied a Convolutional Neural Network (CNN), combined with an attention-enhanced Bidirectional Long Short-Term Memory (BiLSTM) network, to extract the local contexts and long-distance interdependencies from primary sequences. In addition, a multi-task learning strategy was used to predict the secondary structures and the transmembrane helixes. TMPSS was thoroughly trained and tested against a non-redundant independent dataset, where the Q3 secondary structure prediction accuracy achieved 78% in the non-transmembrane region, and the accuracy of the transmembrane region prediction achieved 90%. In sum, our method showcased a unified model for predicting the secondary structure and topology structure of αTMPs by only utilizing features generated from primary sequences and provided a steady and fast prediction, which promisingly improves the structural studies on αTMPs.

Keywords: protein secondary structure, protein topology structure, deep learning, alpha-helical transmembrane proteins, long short-term memory networks

OPEN ACCESS

Edited by:
Zhibin Lv,
University of Electronic Science and Technology of China, China

Reviewed by:
Pu-Feng Du,
Tianjin University, China
Shiwei Sun,
Chinese Academy of Sciences (CAS), China
You Zhou,
Jilin University, China

****Correspondence:***
Han Wang
wangh101@nenu.edu.cn
Guan Ning Lin
nickgnlin@sjtu.edu.cn

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *16 November 2020*
Accepted: *10 December 2020*
Published: *25 January 2021*

Citation:
Liu Z, Gong Y, Bao Y, Guo Y, Wang H and Lin GN (2021) TMPSS: A Deep Learning-Based Predictor for Secondary Structure and Topology Structure Prediction of Alpha-Helical Transmembrane Proteins. Front. Bioeng. Biotechnol. 8:629937. doi: 10.3389/fbioe.2020.629937

INTRODUCTION

Membrane proteins (MPs) are pivotal players in several physiological events, such as signal transduction, neurotransmitter adhesion, ion transport, etc. (Goddard et al., 2015; Roy, 2015). While transmembrane proteins (TMPs), as an essential type of MPs, span the entire biological membrane with segments exposed to both the inside and the outside of the lipid bilayers (Stillwell, 2016). As the major class of TMPs, alpha-helical TMPs are given great pharmacological importance, accounting for about 60% of known drug targets in the current benchmark (Wang et al., 2019). Nevertheless, the difficulties of acquiring their crystal structures always stand in our way due to their low solubilities in the buffers typically used in 2D-PAGE (Butterfield and Boyd-Kimball, 2004; Nugent et al., 2011). All of this is calling for accurate computational predictors.

Predicting alpha-helical TMPs' tertiary structure directly from amino acid sequences has been a challengeable task in computational biology for many years (Yaseen and Li, 2014), but some indirect measures may be worth considering. Since Pauling et al. (1951) performed the first protein secondary structure prediction in 1951, many indicators on the secondary structure level of proteins, such as topology structure (Wang et al., 2019), surface accessibility (Lu et al., 2019), have been demonstrated to be strongly associated with the 3D information of TMPs. Specifically, the secondary structure helps to identify function domains and guides the design of site-specific mutation experiments (Drozdetskiy et al., 2015), whereas the topology structure can help reveal the relative position relationship between TMPs and membranes (Tusnady and Simon, 2001). Generally, the performance of protein secondary structure prediction can be measured by Q3 accuracy in a 3-class classification, i.e., helix (H), strand (E), and coil (C), or Q8 accuracy in an 8-class classification under a more sophisticated evaluation system. Q3 is preferred according to its low cost and close ability in depicting the secondary structure compared with Q8.

Progress in the structure prediction for MPs is slower than that for soluble proteins (Xiao and Shen, 2015). At present, state-of-the-art methods aiming at predicting the secondary structure based on primary sequences, such as SSpro/ACCpro 5 (Magnan and Baldi, 2014), JPred4 (Drozdetskiy et al., 2015), PSIPRED 4 (Buchan and Jones, 2019), and MUFOLD-SSW (Fang et al., 2020), are all trained on soluble protein-specific datasets. However, none of those mentioned methods can simultaneously predict the secondary structure and topology structure of alpha-helical TMPs. More specifically, existing tools could not distinguish transmembrane helices of TMPs from non-transmembrane ones and, in-term, would weaken the TMPs' structure prediction specificity. Another common challenge among the available methods is that features fed into these models are often too miscellaneous, making the model prediction low efficient and even difficult for users to understand. Thus, a more suitable and practical tool for assisting the structure prediction of TMPs is greatly needed.

Deep learning has been employed in several protein sequence classification problems (Lv et al., 2019; Wei et al., 2019; Zeng et al., 2020). Here, we proposed a deep learning-based predictor named TMPSS to predict the secondary structure and topology structure of alpha-helical TMPs simultaneously using amino acid sequences. Equipped with a robust network and carefully screened input features, TMPSS ignored input length restriction and achieved the highest output efficiency compared with other state-of-the-art methods with an acceptable Q3 performance of secondary structure prediction in the full chain (see **Figure 1**). In addition, our TMPSS achieved the Q3 of a whopping 0.97 in the transmembrane region, suggesting that almost all the transmembrane helices were identified. Moreover, TMPSS also significantly outperformed other existing topology structure predictors with the prediction accuracy of 0.90 and the Matthew Correlation Coefficient (MCC) of 0.76 using an independently generated dataset. TMPSS implemented a deep neural network by grouped multiscale Convolutional Neural Networks (CNNs) and stacked attention-enhanced Bidirectional Long Short-Term Memory (BiLSTM) layers for capturing local contexts and global dependencies, respectively. We also utilized the multi-task learning technique to improve prediction performance by considering the mutual effects between different protein properties. We have released TMPSS as a publicly available prediction tool for the community. The pre-trained model and support materials are both available at https://github.com/NENUBioCompute/TMP-SS.

MATERIALS AND METHODS

Benchmark Datasets

As illustrated above, none of the existing secondary structure predictors available today are specific to TMPs. Thus, it is necessary to create unique datasets that contain only alpha-helical TMPs for targeted research. The Protein Data Bank of transmembrane proteins (PDBTM) (Kozma et al., 2012), the first up-to-date and comprehensive TMP selection of the Protein Data Bank (PDB) (Burley et al., 2017), was chosen to construct our datasets. We downloaded 4,336 alpha-helical TMPs from PDBTM (version: 2020-2-7) and removed the chains that contained unknown residues (such as "X") and whose length was <30 residues.

To reduce the redundancy of data and avoid the influence of homology bias (Zou et al., 2020), we utilized CD-HIT (Fu et al., 2012) with a 30% sequence identity cut-off and obtained 911 protein chains. These protein chains were then randomly divided into a training set of 811 chains, a validation set of 50 chains, and a test set (named "TEST50") of 50 chains. Secondary structure labels were obtained by the DSSP program (Kabsch and Sander, 1983) through PDB files, and topology structures were collected from PDBTM. All the experiments were conducted on five-fold cross-validation to gauge its generalization performances (Walsh et al., 2016). The results were used to evaluate our model and compare against other predictors. The overview of AA composition of the training set, validation set, and TEST50 is shown in **Table 1**.

Features and Input Encoding

Features are the key issue for the machine learning tasks (Patil and Chouhan, 2019; Zhang and Liu, 2019). Prediction of alpha-helical TMPs' secondary structure and topology structure at the residue level is formulated as follows: for a given primary protein sequence of an alpha-helical TMP, a sliding window whose length is L residues is used to predict the secondary structure and topology structure of the central residue. For example, if L is 19, each protein will be sliced into fragments of 19 amino acids. Providing valuable input features to deep learning networks is of great importance to make predictions more accurate. Here, we carefully selected two encoding features to represent the protein

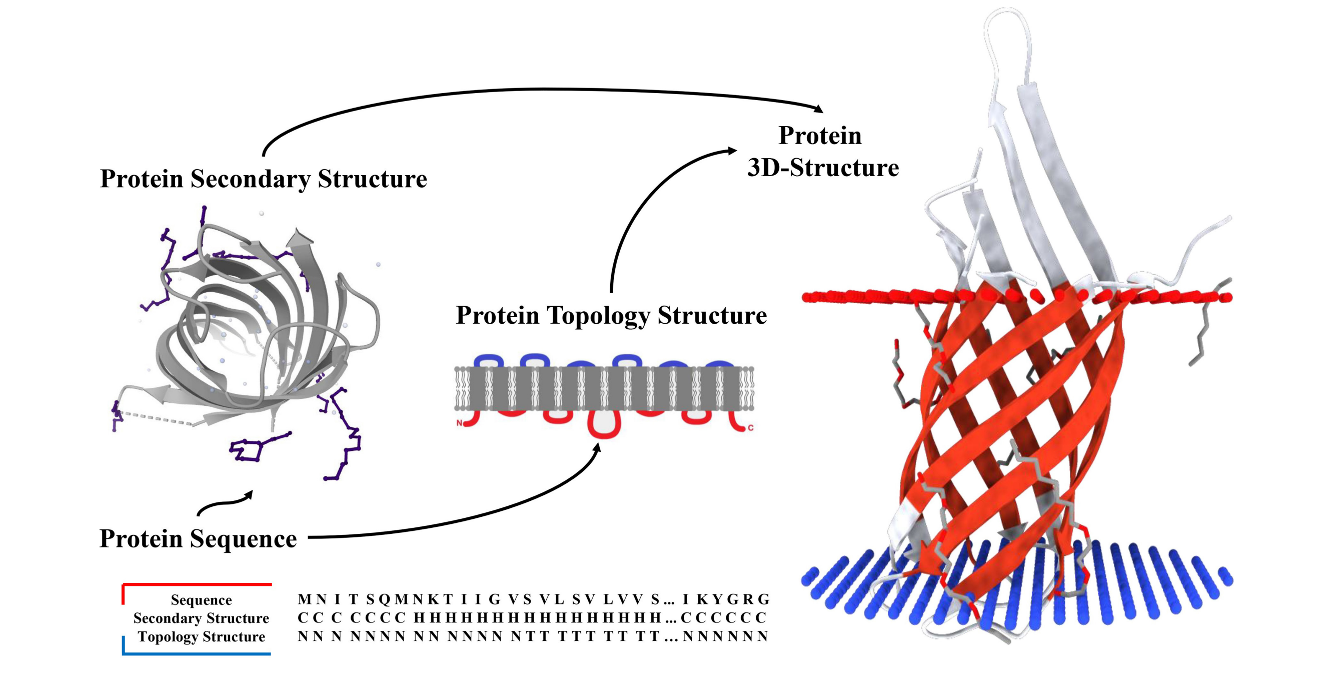

Protein Secondary Structure
Protein 3D-Structure
Protein Topology Structure
N
C
Protein Sequence
Sequence
Secondary Structure
Topology Structure
M N I T S Q M N K T I I G V S V L S V L V V S … I K Y G R G
C C C C C C C H H H H H H H H H H H H H H H H … C C C C C C
N N N N N N N N N N N N N N T T T T T T T T T … N N N N N N

FIGURE 1 | Secondary structure and topology structure prediction of alpha-helical transmembrane proteins.

TABLE 1 | Overview of AA composition of the training set, validation set, and TEST50.

3-State	8-State	Training set		Validation set		TEST50	
Helices	G	60.1%	5,090	59.0%	438	55.1%	317
	H		119,987		7,931		7,897
	I		3,101		254		192
Strands	E	6.3%	12,226	6.5%	853	8.9%	1,240
	B		1,295		103		110
Coils	C	33.5%	34,372	34.5%	2,298	36.0%	2,607
	S		17,861		1,332		1,397
	T		19,195		1,409		1,488

fragment: one-hot code and HHblits profile (Remmert et al., 2012).

The first set came from the protein profiles generated by HHblits, which is faster, almost twice as sensitive, and provides more accurate evolutionary information for protein sequence than PSI-BLAST (Steinegger et al., 2019). We found the best results against the database named uniprot20_2016_02 with three iterations, an E-value threshold of 0.001, and other default settings. The obtained H_{hhm} matrix consisted of 31 dimensions, 30 of which were HMM profile values and one reflected *NoSeq* label (representing a gap) (Fang et al., 2018) at the last column. Each of H_{ij} in the matrix was scaled by a variation of sigmoid function [see Equation (1)], making the distribution of features more uniform and reasonable.

$$f\left(\mathbf{t}\right) = \frac{\mathbf{10}}{\mathbf{1}+e^{-\frac{t}{\mathbf{2000}}}} \tag{1}$$

We then adopted a 21-dimensional matrix O_{onehot} as our second set containing a simple one-hot encoding of 20 positions with one *NoSeq* label. The past research suggested that one-hot encoding was straightforward to generate and has been successfully used in protein structure prediction-associated tasks (Ding and Li, 2015). Therefore, we used 19 dimensional “0” vector with a “1” to represent AA at the index of a particular protein sequence. We mapped each protein fragment sliced by the sliding window with this encoding strategy into an undisturbed coding within local position information.

Model Design

Network Architecture

As a deep learning-based predictor, TMPSS can predict the secondary structure and topology structure of alpha-helical TMPs simultaneously. As we can see in **Figure 2**, the four parts of our model are feature-integration layers for input feature preprocessing, grouped multiscale CNN layers, attention-enhanced BiLSTM layer, and fully-connected layers by two softmax outputs in the end.

Our network's input carried two types of features generated from primary sequences, amino acid features, and profile features. These preprocessed features were fed into a grouped multiscale CNN layer to capture local position information and prevent their mutual interferences at the same time. Then, the merged CNN output flew into two stacked BiLSTM layers, which turned out to be skilled in extracting long-term dependencies and global information (Zhou et al., 2016). We also proposed the attention mechanism as a simple dense layer to help LSTM know which unit's output should be paid more attention. At

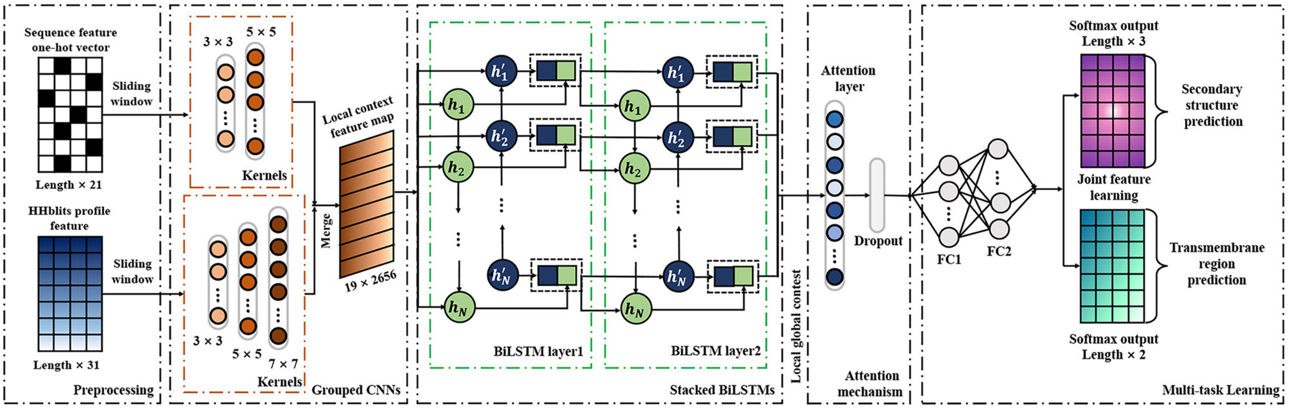


FIGURE 2 | Diagram of TMPSS.

the end of the components mentioned above, there were two fully-connected hidden layers with a softmax-activated output layer, which performed a 3-category secondary structure and 2-category topology structure classification. More details of grouped multiscale CNNs and attention-enhanced BiLSTM are discussed in the **Supplementary Material**.

Implementation Details

Our model was implemented, trained, and tested using the open-source software library Keras (Gulli and Pal, 2017) and Tensorflow (Abadi et al., 2016) on an Nvidia 1080Ti GPU. Main hyperparameters, such as sliding window length, training dropout rate, and number of LSTM units, were explored, and an early stopping strategy and a save-best strategy were adopted (Fang et al., 2018). When the validation loss did not reduce in 10 epochs during training time, the training process would be stopped, and the best model parameters would be saved. In all cases, the weights were initialized by default setting in Keras; the parameters were trained using an Adam optimizer (Bello et al., 2017) to change the learning rate during model training dynamically. Furthermore, batch normalization layers (Ioffe and Szegedy, 2015) and a Dropout layer (Gal et al., 2017) (rate = 0.30) were utilized since they were both skilled in avoiding the network from overfitting and improving the speed of the training process effectively. We set the sliding window's length as 19 residues and put 700 units in each LSTM layer according to the hyperparameter tuning results in this study.

Performance Evaluation

A commonly used evaluation metric for both secondary structure and topology structure prediction based on the residue level is accuracy (ACC), and in particular, Q3 was widely used as a performance metric for 3-category secondary structure prediction (Fang et al., 2017). To quantitatively evaluate the performance of TMPSS and other predictors at the residue level, they were assessed by six measures, including accuracy, recall, precision, specificity, MCC, and F-measure (Tan et al., 2019; Yang et al., 2019; Zhu et al., 2019). The calculation formulas of these evaluation parameters were illustrated as follows:

$$Accuracy = \frac{TN+TP}{TP+FN+FP+TN} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

$$Specificity = \frac{TN}{FP+TN} \tag{5}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{6}$$

$$F-measure = \mathbf{2} \times \frac{Recall \times Precision}{Recall + Precision} \tag{7}$$

where TN, TP, FN, and FP, respectively denoted true negative, true positive, false negative, and false positive samples.

RESULTS

Prediction Performance Analysis at the Residue Level

To evaluate the prediction performance of each category in both two classification tasks at the residue level, we used the confusion matrices (see **Figure 3**), Receiver Operating Characteristic (ROC) curves, and Precision–Recall (PR) curves (see **Figure 4**) to visualize the predict results of TMPSS on TEST50. As illustrated in **Table 1**, TEST50 contains a total of 15,248 residues labeled by "H" (helix), "E" (strand), or "C" (coil) in secondary structure prediction and "T" (transmembrane helix) or "N" (non-transmembrane residue) in topology structure prediction.

Figures 3A,B shows the confusion matrices of secondary structure prediction in the full chain and non-transmembrane region, respectively. As we can see, class "H" was predicted

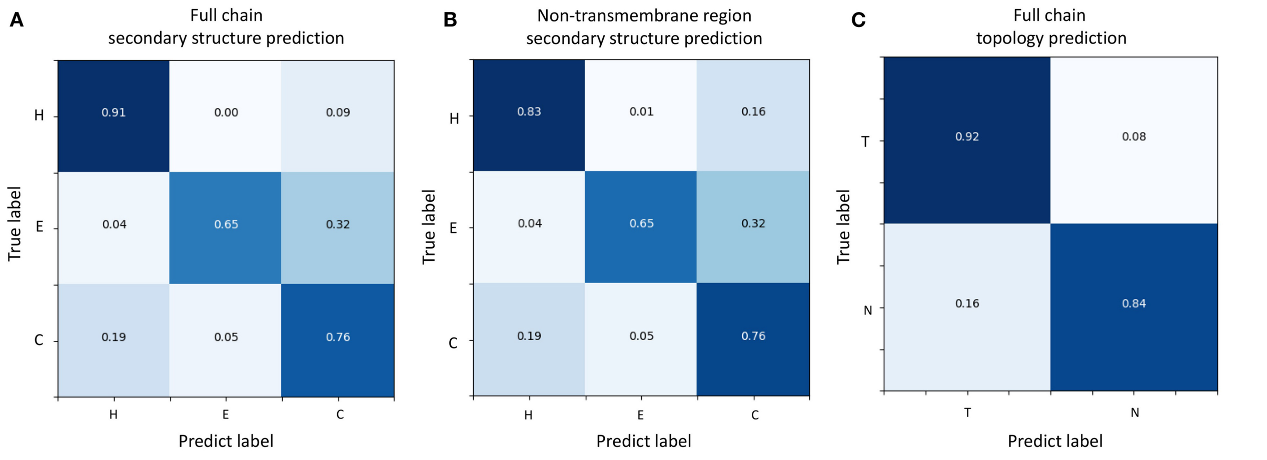


FIGURE 3 | Confusion matrices of TMPSS's prediction performance. **(A)** Confusion matrix of secondary structure prediction in the full chain. **(B)** Confusion matrix of secondary structure prediction in the non-transmembrane region. **(C)** Confusion matrix of topology structure prediction in the full chain.

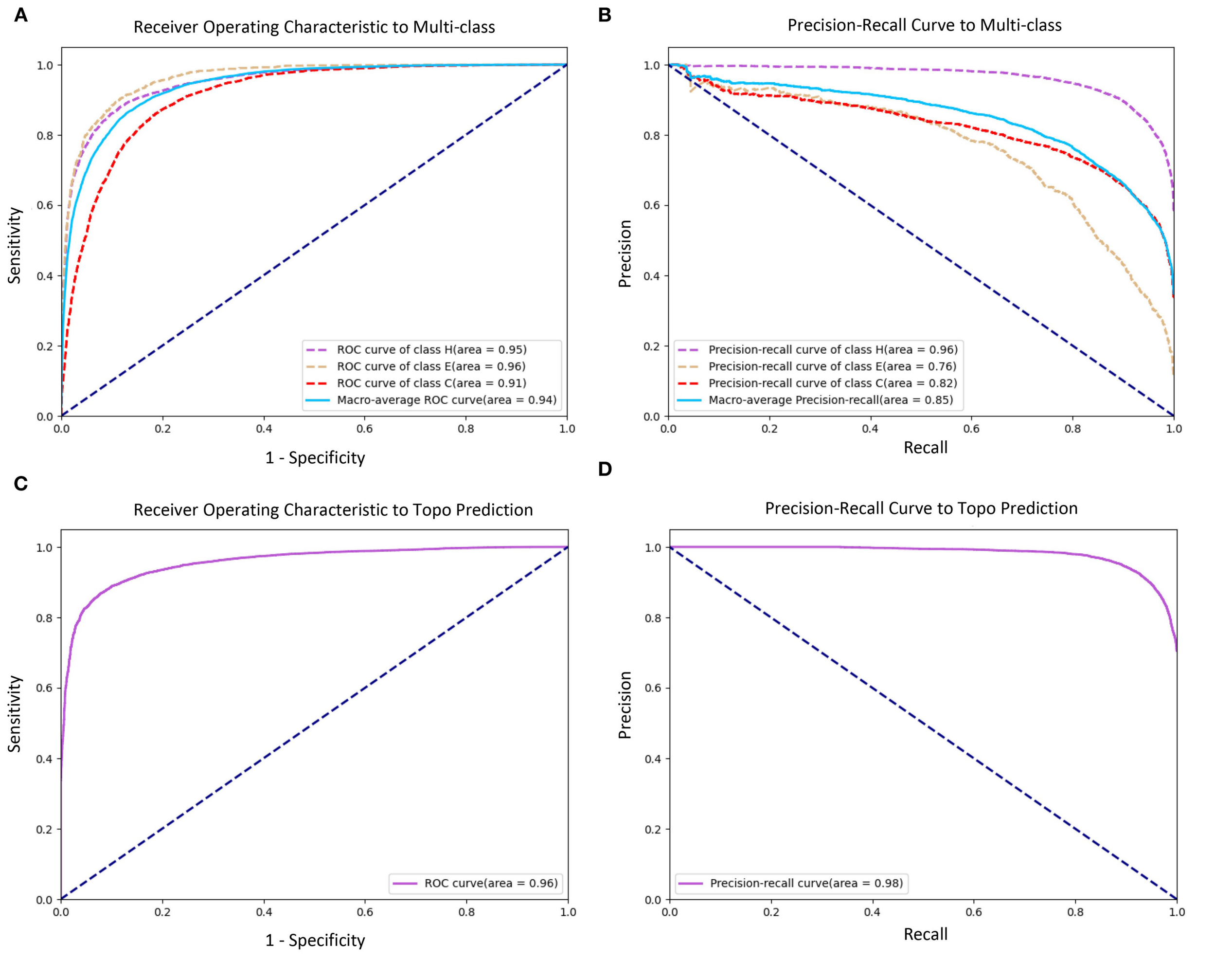


FIGURE 4 | Receiver Operating Characteristic (ROC) and Precision–Recall (PR) curves of prediction performance. **(A)** ROC curve of secondary structure prediction. **(B)** PR curve of secondary structure prediction. **(C)** ROC curve of topology structure prediction. **(D)** PR curve of topology structure prediction.

TABLE 2 | Comparison of TMPSS with previous secondary structure predictors on TEST50 in the full chain.

Method	Class	R	P	S	MCC	F	Full chain SS Q3	Limitation of input length (residues)	Time cost (min)
SSpro5 (with templates)	H	0.908	**0.942**	**0.923**	**0.826**	**0.925**	**0.90**	Limited to 1,500	980
	E	**0.908**	**0.778**	0.975	**0.824**	0.838			
	C	**0.870**	**0.854**	**0.926**	**0.792**	**0.862**			
PSIPRED 4	H	0.907	0.880	0.829	0.741	0.893	0.83	Limited to 1,500	490
	E	0.726	0.735	0.975	0.705	0.731			
	C	0.731	0.770	0.891	0.631	0.750			
RaptorX-Property	H	0.897	0.910	0.877	0.772	0.903	0.85	–	114
	E	0.771	0.761	0.977	0.743	0.766			
	C	0.786	0.770	0.883	0.666	0.778			
Porter 5	H	0.919	0.893	0.849	0.773	0.906	0.85	–	1,035
	E	0.757	0.763	0.977	0.737	0.760			
	C	0.758	0.796	0.903	0.670	0.777			
DeepCNF	H	0.867	0.908	0.879	0.741	0.887	0.83	–	3,000
	E	0.741	0.703	0.970	0.694	0.722			
	C	0.791	0.743	0.864	0.645	0.766			
Spider3	H	0.927	0.883	0.831	0.766	0.904	0.85	–	720
	E	0.751	0.765	0.978	0.734	0.758			
	C	0.737	0.803	0.910	0.662	0.769			
SPOT-1D	H	**0.931**	0.884	0.832	0.772	**0.907**	0.85	Limited to 750	2,030
	E	0.821	0.767	0.976	0.773	0.793			
	C	0.731	0.822	0.921	0.673	0.774			
MUFOLD-SSW	H	0.920	0.884	0.833	0.760	0.902	0.85	Limited to 700	150
	E	0.820	0.743	0.973	0.758	0.779			
	C	0.724	0.815	0.918	0.663	0.767			
JPred4	H	0.830	0.908	0.884	0.706	0.867	0.80	Limited to 800	110
	E	0.664	0.602	0.958	0.595	0.632			
	C	0.772	0.689	0.826	0.583	0.728			
TMPSS	H	0.907	0.888	0.842	0.752	0.897	0.84	–	**96**
	E	0.646	0.764	**0.981**	0.677	0.700			
	C	0.763	0.759	0.880	0.641	0.761			

H, helix (DSSP classes H, G, and I); E, strand (DSSP classes E and B); C, coil (DSSP classes S, T, and blank).
R, Recall; P, Precision; S, Specificity; F, F-measure. Bold fonts represent the best experimental results.

with great precision in different regions of TMPs, but the results of class "E" and class "C" were less satisfactory. A similar experimental phenomenon existed in **Figures 4A,B** simultaneously. Helices account for the largest proportion and make the prediction more significant by considering our dataset's characteristics. The matrices demonstrate that TMPSS did well in both full chain and non-transmembrane region prediction of secondary structure on TEST50, confirming it to be a suitable secondary structure predictor for TMPs.

As for topology structure prediction, TMPSS is also an effective method. The confusion matrix of topology structure prediction in the full chain (see **Figure 3C**) proves that the output results performed well, whether for class "T" or class "N." The ROC and PR curves (see **Figures 4C,D**) also support the above conclusion. After doing a thorough analysis of TMPSS's prediction performance at the residue level on TEST50, it can be seen that TMPSS is a reliable and convenient tool for predicting the secondary structure and topology structure of alpha-helical TMPs synchronously.

Assessment of Multiple Predictors on TEST50

We tested TMPSS against SSpro5 (Magnan and Baldi, 2014) (with templates), PSIPRED 4 (Buchan and Jones, 2019), RaptorX-Property (Wang et al., 2016a), Porter 5 (Torrisi et al., 2019), DeepCNF (Wang et al., 2016b), Spider3 (Heffernan et al., 2017), SPOT-1D (Hanson et al., 2019), MUFOLD-SSW (Fang et al., 2020), and JPred4 (Drozdetskiy et al., 2015) on the TEST50 we created (see **Table 2**). Experimental results illustrated that SSpro5 (with templates) was the most accurate 3-state predictor in our tests on TEST50 in the full chain with a Q3 of 0.90. It might be probably because of the contribution of templates.

TABLE 3 | Comparison of TMPSS with previous secondary structure predictors on TEST50 in the different transmembrane regions.

Method	Trans SS Q3	Non-trans SS Q3
SSpro5 (with templates)	0.90	**0.89**
PSIPRED 4	0.94	0.79
RaptorX-Property	0.95	0.80
Porter 5	0.95	0.81
DeepCNF	0.91	0.80
Spider3	0.95	0.80
SPOT-1D	0.95	0.81
MUFOLD-SSW	0.94	0.81
JPred4	0.90	0.75
TMPSS	**0.97**	0.78

Trans, transmembrane region; Non-trans, non-transmembrane region. Bold fonts represent the best experimental results.

However, apart from SSpro5 (with templates), the remaining servers performed similarly with the maximum Q3 deviation of 0.02, and some servers, such as JPred4, even performed worse. Many methods refused to accept sequences of more than a certain length. By comparison, TMPSS was user-friendly with no length limitation of input and had the highest output efficiency among the existing methods with an acceptable Q3 of 0.84 in the full chain.

It is worth emphasizing that this comparison shown in **Table 2** is “unfair” for our experimental tool. Firstly, the existing secondary structure predictors cannot distinguish the transmembrane “H’s” from non-transmembrane “H’s”, whereas ours can. Secondly, some tools, such as SSpro5, uses templates, which cannot be found when making predictions about unknown structural sequences and not recommended to use under normal circumstances.

However, the tools suitable for water-soluble proteins may not be suitable for handling the residues in the transmembrane region of TMPs since they cannot distinguish transmembrane helices from non-transmembrane helices. To assess different servers’ secondary structure prediction ability in the different transmembrane regions, we calculated the precision of both transmembrane and non-transmembrane residues and listed the results in **Table 3**. As expected, TMPSS achieved the best Q3 performance among all exemplified servers in the transmembrane region, which signified that almost all the transmembrane helices were identified by our method.

As for topology prediction, we compared TMPSS to state-of-the-art topology predictors, including HMMTOP 2 (Tusnady and Simon, 2001), OCTOPUS (Viklund and Elofsson, 2008), TOPCONS (Tsirigos et al., 2015), Philius (Reynolds et al., 2008), PolyPhobius (Jones, 2007), SCAMPI (Bernsel et al., 2008), and SPOCTOPUS (Viklund et al., 2008). As illustrated in **Table 4**, TMPSS obtains the best ACC (= 0.90) and MCC (= 0.76) performance on TEST50 in the full chain among the listed methods. The most probable cause is that the joint feature learning helped two prediction tasks promote each other. According to this, the deep convolutional BiLSTM extracted the most effective information though there are only two features exploited.

TABLE 4 | Comparison of TMPSS with state-of-the-art topology predictors on TEST50 in the full chain.

Method	ACC	MCC
HMMTOP 2	0.84	0.64
OCTOPUS	0.87	0.71
TOPCONS	0.88	0.72
Philius	0.87	0.71
PolyPhobius	0.88	0.72
SCAMPI	0.87	0.70
SPOCTOPUS	0.87	0.71
TMPSS	**0.90**	**0.76**

Bold fonts represent the best experimental results.

TABLE 5 | Effect of loss weight during multi-task learning.

Loss weight ($\lambda_1 : \lambda_2$)	SS Q3	Topo ACC
1:0.1	0.832	0.887
1:0.3	0.833	0.892
1:0.5	**0.835**	**0.896**
1:0.7	0.825	0.892
1:1	0.830	0.894
1:5	0.811	0.889
1:10	0.794	0.892

Bold fonts represent the best experimental results.

Multi-Task Learning

Secondary structure prediction and topology structure prediction of alpha-helical TMPs are highly related tasks since the residues labeled “T” (transmembrane helix) in topology structure prediction also have the label of “H” (helix) in secondary structure prediction (Chen et al., 2002). Therefore, we put these two tasks together to support multi-task learning (Zhang and Yeung, 2012) and generated a 3-class secondary structure and a 2-class topology structure simultaneously. With the help of multi-task learning, our model’s computational complexity was significantly reduced compared with other methods based on cascaded deep learning networks. The joint loss function could be formulated as follows:

$$L\left(\{s_i, t_i\}\right) = \frac{\lambda_1}{N} \sum L_s\left(s_i, s_i{}^*\right) + \frac{\lambda_2}{N} \sum L_t\left(t_i, t_i{}^*\right) \tag{8}$$

where $L_s\left(s_i, s_i{}^*\right) = -s_i{}^* log(s_i)$ and $L_t\left(t_i, t_i{}^*\right) = -[t_i{}^* log(t_i) + (1 - t_i{}^*) log(1 - t_i)]$ are respective loss functions for secondary structure and topology structure prediction, s_i and t_i are predicted probabilities (softmax output) of secondary structure labels and topology structure labels, respectively, $s_i{}^*$ and $t_i{}^*$ are ground-truth labels of secondary structure and topology structure, respectively, λ_1 and λ_2 are loss weight of combined loss function, and N is the total number of residues. **Table 5** shows the effect of different loss weights ($\lambda_1 : \lambda_2$) during multi-task learning on the validation dataset, and we set $\lambda_1 = 1$, $\lambda_2 = 0.5$

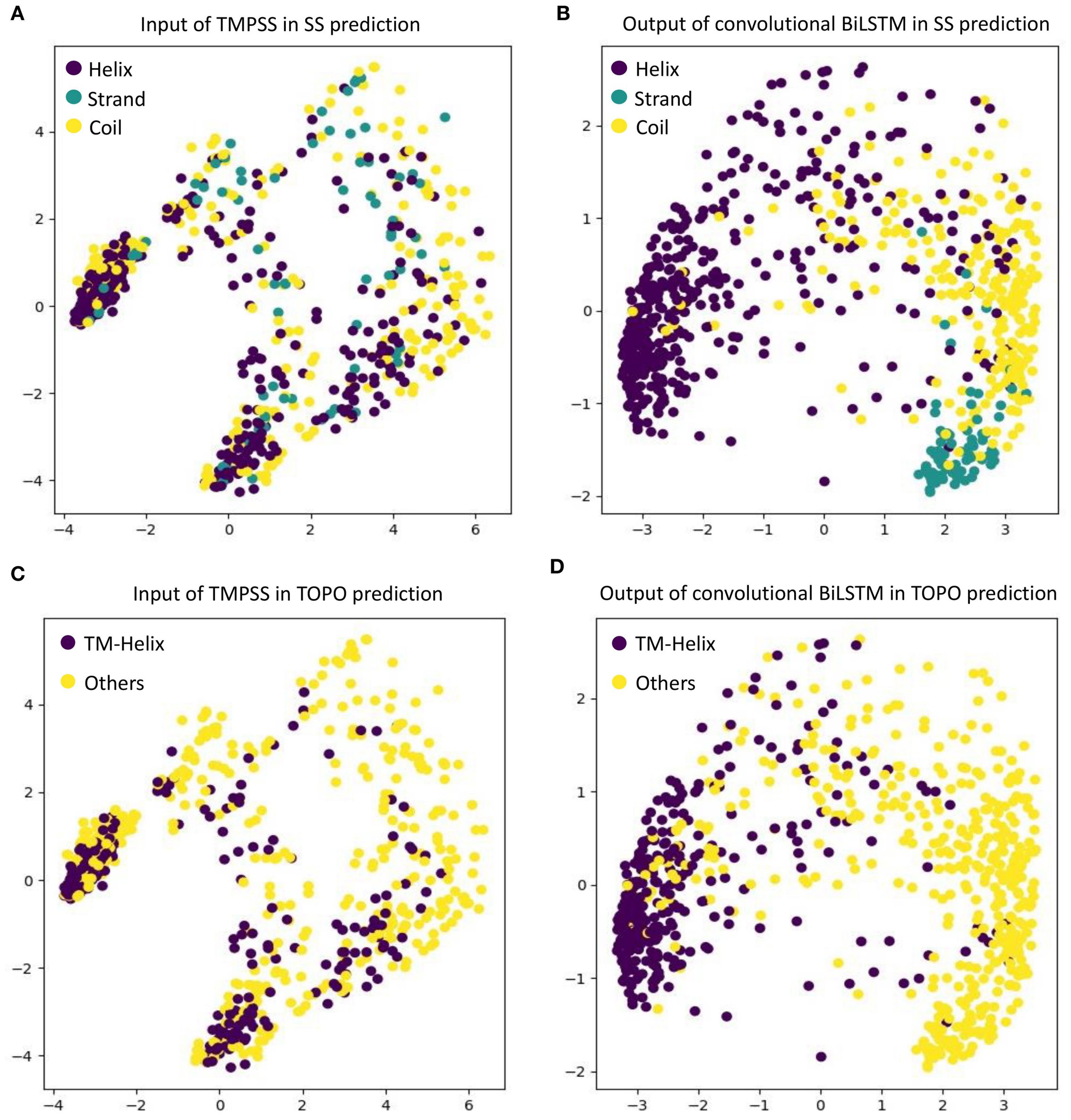


FIGURE 5 | Visualize the input features and the features learned by convolutional BiLSTM, respectively, using PCA. **(A)** Input of TMPSS in SS prediction. **(B)** Output of convolutional BiLSTM in SS prediction. **(C)** Input of TMPSS in TOPO prediction. **(D)** Output of convolutional BiLSTM in TOPO prediction.

for balancing two joint feature learning tasks and regularization terms in the end.

Visualization of the Features Learnt by Convolutional BiLSTM

As an automatic feature extraction process, deep learning can learn high-level abstract features from original inputs (Farias et al., 2016). To further explore the effectiveness of convolutional BiLSTM, Principal Component Analysis (PCA) (Shlens, 2014) was utilized to visualize the input features and each LSTM unit's output in the last bidirectional layer with TEST50. **Figure 5** shows the PCA scatter diagrams before and after TEST50 was fed into our network, respectively.

TABLE 6 | Effect of different combination ways of the attention mechanism on TEST50.

Model	SS Q3	Topo ACC
Attention with multiscale CNNs	0.826	0.893
Attention with BiLSTM	**0.835**	**0.896**
Attention with dropout	0.742	0.866

Bold fonts represent the best experimental results.

As described earlier, the input data had 52 features (i.e., 52 dimensions). PCA reduced the input features' dimensionality to two principal dimensions and visualized it. As we can see in **Figures 5A,C**, no clear cluster can be found. However, after feeding the data into the convolutional BiLSTM that contains 1,400 dimensions (twice of the unit number in a simple LSTM) at the top layer, the data points showed apparent clustering tendency (see **Figures 5B,D**). This visualization experiment strongly proved the feature extraction efficiency of the convolutional BiLSTM.

It is worth mentioning that since multi-task joint feature learning was performed in our network, the label-based visualization results also revealed the internal relation between secondary structure prediction and topology structure prediction. We found that the points representing "helices" of secondary structure and the ones representing "transmembrane helices" of topology structure have almost completely overlapping distributions under different label-orientated predictions. This experimental phenomenon also directly confirmed the strong correlation between the two prediction tasks and the necessity and effectiveness of multi-task learning.

TABLE 7 | An ablation study on TEST50.

Model	SS Q3	Topo ACC
Without multiscale CNNs	0.832	0.895
Without BiLSTM layers	0.759	0.743
Without multi-task learning	0.825	0.891
Without attention mechanism	0.828	0.892
TMPSS	**0.835**	**0.896**

Bold fonts represent the best experimental results.

More results, such as the prediction performance analysis at the residue level, feature analysis, implementation details of multi-task learning, implementation details of attention mechanism, and an ablation study, can be found in the **Supplementary Material**.

Attention Mechanism

The attention mechanism can stimulate the model extracting features more effectively, speeding up reaching or even improving the best performance of prediction (Choi et al., 2016). To verify the effect of various binding ways of attention mechanism, which acted as a simple full-connect layer in our model, we combined it with different network layers, and the results are shown in **Table 6**. It can be seen that when we attached an attention layer to BiLSTM layers, the prediction results (SS Q3 = 0.835 and Topo ACC = 0.896) were better than doing the same thing to multiscale CNNs or the Dropout layer as expected. One reason could be that the attention mechanism enhanced the process of feature extraction. Another reason could be that BiLSTM layers just learned the most abundant contextual features, making it achieve the best effect when combining attention layer with BiLSTM layers.

Ablation Study

To discover whether a certain component of our proposed method was vital or necessary, we carried out an ablation study by removing some network elements in this section. The experiments performed in our ablation study shared the same

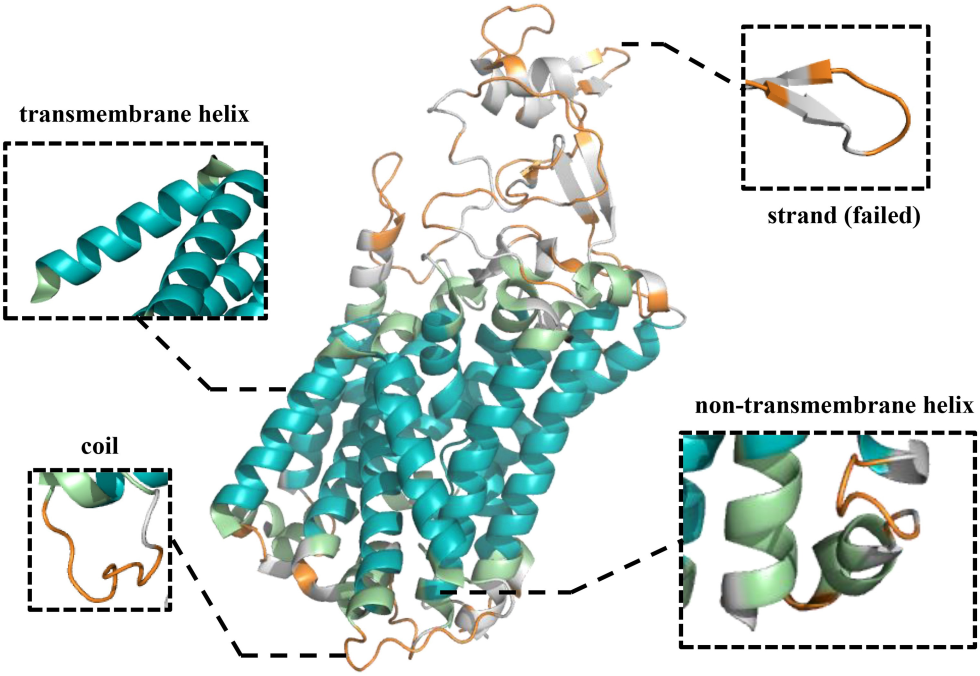


FIGURE 6 | Visualization of secondary structure and topology structure prediction results generated by TMPSS with PyMOL: take 6KKT_A as an example.

features and hyperparameters. From the results on TEST50 presented in **Table 7**, we found that those BiLSTM layers were the most contributing and effective component in our model since the Q3 accuracy of secondary structure prediction dropped to 75.9% when we roughly removed this part from the network. Multiscale CNNs were also essential for good performance as they were particularly good at dealing with local information of protein sequences. Furthermore, multi-task learning and attention mechanism were necessary at the same time because their application made contributions to the robustness of our method with the proof of study results.

Case Study

To further demonstrate the effectiveness of TMPSS on predicting the secondary structure and topology structure of alpha-helical TMPs, we randomly took 6KKT_A as an example of our case study. 6KKT is a kind of transport protein of *Homo sapiens* released on 2019-10-23 that plays vital roles in cell volume regulation, ion transport, and salt reabsorption in the kidney (Liu et al., 2019). The prediction result of TMPSS is visualized in **Figure 6** using PyMOL (DeLano, 2002).

As can be seen, our model correctly identified the helices in the transmembrane region (colored blue) and the non-transmembrane region (colored green). Additionally, most of the coils in the non-transmembrane region (colored orange) were also successfully distinguished.

CONCLUSION

In this study, we proposed a deep learning-based predictor, TMPSS, to predict the secondary structure and topology structure of alpha-helical TMPs from primary sequences. TMPSS's Q3 accuracy of secondary structure prediction in the full chain performed on par with the state-of-the-art methods statistically, and our model had the highest output efficiency with no length restriction of input at the same time. Moreover, our method achieved the best Q3 performance in the transmembrane region and significantly outperformed other topology structure predictors on the independent dataset TEST50.

TMPSS applied a deep learning network with grouped multiscale CNNs and stacked attention-enhanced BiLSTM layers for capturing local and global contexts. Multi-task learning was exploited to improve prediction performance and reduce our method's computational expense by considering the interactions between different protein properties. A series of visualization experiments and comparative tests was taken to verify the validity of the model components mentioned above.

Furthermore, we implemented TMPSS as a publicly available predictor for the research community. The pre-trained model and the datasets we used in this paper could be downloaded at https://github.com/NENUBioCompute/TMP-SS. Finally, we sincerely hope that the predictor and the support materials we released in this study will help the researchers who need them.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Materials**, further inquiries can be directed to the corresponding authors.

AUTHOR CONTRIBUTIONS

ZL, YGo, and YB conceived the idea of this research, collected the data, implemented the predictor, and wrote the manuscript. ZL and YGu tuned the model and tested the predictor. HW and GL supervised the research and reviewed the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by grants from the National Natural Science Foundation of China (nos. 81671328, 81971292), the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning (no. 1610000043), the Innovation Research Plan supported by Shanghai Municipal Education Commission (ZXWF082101), the Jilin Scientific and Technological Development Program (no. 20180414006GH), and the Fundamental Research Funds for the Central Universities (nos. 2412019FZ052, 2412019FZ048).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fbioe.2020.629937/full#supplementary-material

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). "Tensorflow: a system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation (OSDI´16)* (Savannah, GA), 265–283.

Bello, I., Zoph, B., Vasudevan, V., and Le, Q. V. (2017). "Neural optimizer search with reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70: JMLR. org* (Sydney, NSW), 459–468.

Bernsel, A., Viklund, H., Falk, J., Lindahl, E., von Heijne, G., and Elofsson, A. (2008). Prediction of membrane-protein topology from first principles. *Proc. Natl. Acad. Sci. U.S.A.* 105, 7177–7181. doi: 10.1073/pnas.0711151105

Buchan, D. W., and Jones, D. T. (2019). The PSIPRED protein analysis workbench: 20 years on. *Nucleic Acids Res.* 47, W402–W407. doi: 10.1093/nar/gkz297

Burley, S. K., Berman, H. M., Kleywegt, G. J., Markley, J. L., Nakamura, H., and Velankar, S. (2017). Protein Data Bank (PDB): the single global macromolecular structure archive. *Methods Mol. Biol.* 1607, 627–641. doi: 10.1007/978-1-4939-7000-1_26

Butterfield, D. A., and Boyd-Kimball, D. (2004). Proteomics analysis in Alzheimer's disease: new insights into mechanisms of neurodegeneration. *Int. Rev. Neurobiol.* 61, 159–188. doi: 10.1016/S0074-7742(04)61007-5

Chen, C. P., Kernytsky, A., and Rost, B. (2002). Transmembrane helix predictions revisited. *Protein Sci.* 11, 2774–2791. doi: 10.1110/ps.0214502

Choi, E., Bahadori, M. T., Sun, J., Kulas, J., Schuetz, A., and Stewart, W. (2016). Retain: an interpretable predictive model for healthcare using reverse time attention mechanism. *Adv. Neural Inf. Process. Syst.* 3504–3512.

DeLano, W. L. (2002). Pymol: an open-source molecular graphics tool. *CCP4 Newslett. Protein Crystallogr.* 40, 82–92.

Ding, H., and Li, D. (2015). Identification of mitochondrial proteins of malaria parasite using analysis of variance. *Amino Acids* 47, 329–333. doi: 10.1007/s00726-014-1862-4

Drozdetskiy, A., Cole, C., Procter, J., and Barton, G. J. (2015). JPred4: a protein secondary structure prediction server. *Nucleic Acids Res.* 43, W389–W394. doi: 10.1093/nar/gkv332

Fang, C., Li, Z., Xu, D., and Shang, Y. (2020). MUFold-SSW: a new web server for predicting protein secondary structures, torsion angles, and turns. *Bioinformatics* 36, 1293–1295. doi: 10.1093/bioinformatics/btz712

Fang, C., Shang, Y., and Xu, D. (2017). MUFold-SS: Protein Secondary Structure Prediction Using Deep Inception-Inside-Inception Networks. *arXiv preprint arXiv:1709.06165*.

Fang, C., Shang, Y., and Xu, D. (2018). Improving protein gamma-turn prediction using inception capsule networks. *Sci. Rep.* 8, 1–12. doi: 10.1038/s41598-018-34114-2

Farias, G., Dormido-Canto, S., Vega, J., Ratt,á, G., Vargas, H., Hermosilla, G., et al. (2016). Automatic feature extraction in large fusion databases by using deep learning approach. *Fusion Eng. Des.* 112, 979–983. doi: 10.1016/j.fusengdes.2016.06.016

Fu, L., Niu, B., Zhu, Z., Wu, S., and Li, W. (2012). CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* 28, 3150–3152. doi: 10.1093/bioinformatics/bts565

Gal, Y., Hron, J., and Kendall, A. (2017). Concrete dropout. *Adv. Neural Inf. Process. Syst.* 3581–3590.

Goddard, A. D., Dijkman, P. M., Adamson, R. J., dos Reis, R. I., and Watts, A. (2015). Reconstitution of membrane proteins: a GPCR as an example. *Methods Enzymol.* 556, 405–424. doi: 10.1016/bs.mie.2015.01.004

Gulli, A., and Pal, S. (2017). *Deep Learning With Keras.* Birmingham: Packt Publishing Ltd.

Hanson, J., Paliwal, K., Litfin, T., Yang, Y., and Zhou, Y. (2019). Improving prediction of protein secondary structure, backbone angles, solvent accessibility, and contact numbers by using predicted contact maps and an ensemble of recurrent and residual convolutional neural networks. *Bioinformatics* 35, 2403–2410. doi: 10.1093/bioinformatics/bty1006

Heffernan, R., Yang, Y., Paliwal, K., and Zhou, Y. (2017). Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers, and solvent accessibility. *Bioinformatics* 33, 2842–2849. doi: 10.1093/bioinformatics/btx218

Ioffe, S., and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprint arXiv:1502.03167.*

Jones, D. T. (2007). Improving the accuracy of transmembrane protein topology prediction using evolutionary information. *Bioinformatics* 23, 538–544. doi: 10.1093/bioinformatics/btl677

Kabsch, W., and Sander, C. (1983). DSSP: definition of secondary structure of proteins given a set of 3D coordinates. *Biopolymers* 22, 2577–2637. doi: 10.1002/bip.360221211

Kozma, D., Simon, I., and Tusnady, G. E. (2012). PDBTM: protein data bank of transmembrane proteins after 8 years. *Nucleic Acids Res.* 41, D524–D529. doi: 10.1093/nar/gks1169

Liu, S., Chang, S., Han, B., Xu, L., Zhang, M., Zhao, C., et al. (2019). Cryo-EM structures of the human cation-chloride cotransporter KCC1. *Science* 366, 505–508. doi: 10.1126/science.aay3129

Lu, C., Liu, Z., Kan, B., Gong, Y., Ma, Z., and Wang, H. (2019). TMP-SSurface: a deep learning-based predictor for surface accessibility of transmembrane protein residues. *Crystals* 9:640. doi: 10.3390/cryst9120640

Lv, Z. B., Ao, C. Y., and Zou, Q. (2019). Protein function prediction: from traditional classifier to deep learning. *Proteomics* 19:e1900119. doi: 10.1002/pmic.201900119

Magnan, C. N., and Baldi, P. (2014). SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning, and structural similarity. *Bioinformatics* 30, 2592–2597. doi: 10.1093/bioinformatics/btu352

Nugent, T., Ward, S., and Jones, D. T. (2011). The MEMPACK alpha-helical transmembrane protein structure prediction server. *Bioinformatics* 27, 1438–1439. doi: 10.1093/bioinformatics/btr096

Patil, K., and Chouhan, U. (2019). Relevance of machine learning techniques and various protein features in protein fold classification: a review. *Curr. Bioinform.* 14, 688–697. doi: 10.2174/1574893614666190204154038

Pauling, L., Corey, R. B., and Branson, H. R. (1951). The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Natl. Acad. Sci. U.S.A.* 37, 205–211. doi: 10.1073/pnas.37.4.205

Remmert, M., Biegert, A., Hauser, A., and Söding, J. (2012). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Methods* 9, 173–175. doi: 10.1038/nmeth.1818

Reynolds, S. M., Käll, L., Riffle, M. E., Bilmes, J. A., and Noble, W. S. (2008). Transmembrane topology and signal peptide prediction using dynamic bayesian networks. *PLoS Comput. Biol.* 4:e1000213. doi: 10.1371/journal.pcbi.1000213

Roy, A. (2015). Membrane preparation and solubilization. *Methods Enzymol.* 557, 45–56. doi: 10.1016/bs.mie.2014.11.044

Shlens, J. (2014). A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100.*

Steinegger, M., Meier, M., Mirdita, M., Vöhringer, H., Haunsberger, S. J., and Söding, J. (2019). HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinform.* 20:473. doi: 10.1186/s12859-019-3019-7

Stillwell, W. (2016). *An Introduction to Biological Membranes: Composition, Structure, and Function*. Amsterdam: Elsevier.

Tan, J.-X., Li, S.-H., Zhang, Z.-M., Chen, C.-X., Chen, W., Tang, H., et al. (2019). Identification of hormone binding proteins based on machine learning methods. *Math. Biosci. Eng.* 16, 2466–2480. doi: 10.3934/mbe.2019123

Torrisi, M., Kaleel, M., and Pollastri, G. (2019). Deeper profiles and cascaded recurrent and convolutional neural networks for state-of-the-art protein secondary structure prediction. *Sci. Rep.* 9, 1–12. doi: 10.1038/s41598-019-48786-x

Tsirigos, K. D., Peters, C., Shu, N., Käll, L., and Elofsson, A. (2015). The TOPCONS web server for consensus prediction of membrane protein topology and signal peptides. *Nucleic Acids Res.* 43, W401–W407. doi: 10.1093/nar/gkv485

Tusnady, G. E., and Simon, I. (2001). The HMMTOP transmembrane topology prediction server. *Bioinformatics* 17, 849–850. doi: 10.1093/bioinformatics/17.9.849

Viklund, H., Bernsel, A., Skwark, M., and Elofsson, A. (2008). SPOCTOPUS: a combined predictor of signal peptides and membrane protein topology. *Bioinformatics* 24, 2928–2929. doi: 10.1093/bioinformatics/btn550

Viklund, H., and Elofsson, A. (2008). OCTOPUS: improving topology prediction by two-track ANN-based preference scores and an extended topological grammar. *Bioinformatics* 24, 1662–1668. doi: 10.1093/bioinformatics/btn221

Walsh, I., Pollastri, G., and Tosatto, S. C. (2016). Correct machine learning on protein sequences: a peer-reviewing perspective. *Brief. Bioinform.* 17, 831–840. doi: 10.1093/bib/bbv082

Wang, H., Yang, Y., Yu, J., Wang, X., Zhao, D., Xu, D., et al. (2019). "DMCTOP: topology prediction of alpha-helical transmembrane protein based on deep multi-scale convolutional neural network," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (San Diego, CA: IEEE), 36–43.

Wang, S., Li, W., Liu, S., and Xu, J. (2016a). RaptorX-Property: a web server for protein structure property prediction. *Nucleic Acids Res.* 44, W430–W435. doi: 10.1093/nar/gkw306

Wang, S., Peng, J., Ma, J., and Xu, J. (2016b). Protein secondary structure prediction using deep convolutional neural fields. *Sci. Rep.* 6, 1–11. doi: 10.1038/srep18962

Wei, L., Su, R., Wang, B., Li, X., Zou, Q., and Gao, X. (2019). Integration of deep feature representations and handcrafted features to improve the prediction of N 6-methyladenosine sites. *Neurocomputing* 324, 3–9. doi: 10.1016/j.neucom.2018.04.082

Xiao, F., and Shen, H.-B. (2015). Prediction enhancement of residue real-value relative accessible surface area in transmembrane helical proteins by solving the output preference problem of machine learning-based predictors. *J. Chem. Inf. Model.* 55, 2464–2474. doi: 10.1021/acs.jcim.5b00246

Yang, W., Zhu, X.-J., Huang, J., Ding, H., and Lin, H. (2019). A brief survey of machine learning methods in protein sub-Golgi localization. *Curr. Bioinform.* 14, 234–240. doi: 10.2174/1574893613666181113131415

Yaseen, A., and Li, Y. (2014). Context-based features enhance protein secondary structure prediction accuracy. *J. Chem. Inf. Model.* 54, 992–1002. doi: 10.1021/ci400647u

Zeng, X., Zhong, Y., Lin, W., and Zou, Q. (2020). Predicting disease-associated circular RNAs using deep forests combined with positive-unlabeled learning methods. *Brief. Bioinform.* 21, 1425–1436. doi: 10.1093/bib/bbz080

Zhang, J., and Liu, B. (2019). A review on the recent developments of sequence-based protein feature extraction methods. *Curr. Bioinform.* 14, 190–199. doi: 10.2174/1574893614666181212102749

Zhang, Y., and Yeung, D.-Y. (2012). A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., et al. (2016). "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (volume 2: Short papers)* (Berlin), 207–212.

Zhu, X.-J., Feng, C.-Q., Lai, H.-Y., Chen, W., and Hao, L. (2019). Predicting protein structural classes for low-similarity sequences by evaluating different features. *Knowl. Based Syst.* 163, 787–793. doi: 10.1016/j.knosys.2018.10.007

Zou, Q., Lin, G., Jiang, X., Liu, X., and Zeng, X. (2020). Sequence clustering in bioinformatics: an empirical study. *Brief. Bioinform.* 21, 1–10.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The reviewer P-FD declared a shared affiliation, with no collaboration, with the author YGo to the handling editor at the time of the review.

frontiers
in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 25 February 2021
doi: 10.3389/fbioe.2021.647113

Prediction of lncRNA–Protein Interactions via the Multiple Information Integration

*Yifan Chen[1,2], Xiangzheng Fu[1], Zejun Li[2], Li Peng[3] and Linlin Zhuo[4]**

[1] College of Information Science and Engineering, Hunan University, Changsha, China, [2] School of Computer and Information Science, Hunan Institute of Technology, Hengyang, China, [3] College of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China, [4] Department of Mathematics and Information Engineering, Wenzhou University Oujiang College, Wenzhou, China

The long non-coding RNA (lncRNA)–protein interaction plays an important role in the post-transcriptional gene regulation, such as RNA splicing, translation, signaling, and the development of complex diseases. The related research on the prediction of lncRNA–protein interaction relationship is beneficial in the excavation and the discovery of the mechanism of lncRNA function and action occurrence, which are important. Traditional experimental methods for detecting lncRNA–protein interactions are expensive and time-consuming. Therefore, computational methods provide many effective strategies to deal with this problem. In recent years, most computational methods only use the information of the lncRNA–lncRNA or the protein–protein similarity and cannot fully capture all features to identify their interactions. In this paper, we propose a novel computational model for the lncRNA–protein prediction on the basis of machine learning methods. First, a feature method is proposed for representing the information of the network topological properties of lncRNA and protein interactions. The basic composition feature information and evolutionary information based on protein, the lncRNA sequence feature information, and the lncRNA expression profile information are extracted. Finally, the above feature information is fused, and the optimized feature vector is used with the recursive feature elimination algorithm. The optimized feature vectors are input to the support vector machine (SVM) model. Experimental results show that the proposed method has good effectiveness and accuracy in the lncRNA–protein interaction prediction.

Keywords: feature representation, mutual information, structure analysis, support vector machine, lncRNA protein interactions

OPEN ACCESS

Edited by:
Zhibin Lv,
University of Electronic Science and Technology of China, China

Reviewed by:
Qiu Xiao,
Hunan Normal University, China
Wei Zhang,
Changsha University, China

****Correspondence:***
Linlin Zhuo
zhuoninnin@163.com

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *29 December 2020*
Accepted: *19 January 2021*
Published: *25 February 2021*

Citation:
Chen Y, Fu X, Li Z, Peng L and Zhuo L (2021) Prediction of lncRNA–Protein Interactions via the Multiple Information Integration. Front. Bioeng. Biotechnol. 9:647113. doi: 10.3389/fbioe.2021.647113

INTRODUCTION

Long non-coding RNA (lncRNA)–protein interactions play an important role in the post-transcriptional gene regulation, polyadenylation, splicing, and translation, and predicting lncRNA–protein interactions helps to understand lncRNA-related activities (Mittal et al., 2009; Ray et al., 2013). With the rapid advancement of high-throughput technologies and the rapid increase of lncRNA and protein sequence data, predicting lncRNA–protein interactions by traditional biological experimental approaches, such as RNA-pulldown, RNA immunoprecipitation, and other biological experiments, is expensive and time-consuming. In recent years, computational methods, especially machine learning methods, have been widely used in the field of bioinformatics. For example, Link prediction paradigms have been used to predict drug targets

(Munir et al., 2019; Srivastava et al., 2019; Zeng et al., 2019, 2020; Ru et al., 2020; Wang et al., 2020), enhancer promoter interactions (Hong et al., 2019; Cai et al., 2020a), disease genes (Zeng et al., 2017a; Ji et al., 2019; Kuang et al., 2019; Wang et al., 2019; Peng et al., 2020), link prediction (Xiao et al., 2018, 2019, 2020), circular RNAs (Zeng et al., 2017b; Xiao et al., 2019), microRNAs (miRNAs) (Xiao et al., 2018, 2020; Zeng et al., 2018; Hajieghrari et al., 2019; Jeyaram et al., 2019; Zhang X. et al., 2019), and peptide recognition (Bai et al., 2019; Cai et al., 2020b; Fu et al., 2020; Zhang and Zou, 2020). In addition, computational intelligence such as evolutionary algorithms (Song et al., 2020a,b) and unsupervised learning (Lambrou et al., 2019; Noureen et al., 2019; Zhang L. et al., 2019; Zou et al., 2020) can be applied to the field of bioinformatics. Given the efficient performance of machine learning methods in predicting lncRNA–protein interactions, the number of researchers considering machine learning methods as the first choice for predicting lncRNA–protein interactions have been increasing.

The general process of machine learning methods for predicting lncRNA–protein interactions is as follows. First, raw lncRNA and protein data are mined and analyzed separately to extract the characteristic information of lncRNA and protein. Algorithms are then designed to compute the lncRNA–protein interactions and obtain their relationships. Finally, prediction results are verified and can be used to guide biological experiments in reverse, which can reduce the cost of biological experiments and improve the efficiency of research. Currently, machine learning-based methods for predicting lncRNA–protein interactions can be divided into two main categories.

(1) Construction of prediction models on the basis of lncRNA and protein features. The feature information of lncRNA and protein can be extracted using feature extraction methods based on sequence information, structure, and various physicochemical properties, which are fused to construct feature vectors. Feature vectors are fed into machine learning classification algorithms to construct prediction models for lncRNA–protein interaction relationships. Bellucci et al. (2011) have proposed the catRAPID model for predicting lncRNA–protein interactions, which combines the protein molecular secondary structure and the position information and extracts and inputs more than 100 dimensions of feature information from protein and non-coding RNA into the random forest (RF) and the support vector machine (SVM) to train the prediction model. Muppirala et al. (2011) have developed the RPISeq method, which utilizes only lncRNA and protein sequence information and uses SVM and RF classifiers to construct a model for the prediction of lncRNA–protein association interactions. Wang et al. (2013) have applied the plain Bayesian to construct prediction models for predicting lncRNA–protein interactions on the basis of the study of Lu et al. (2013) have proposed a method called the lncPro, which extracts amino acid and nucleotide sequence information and applies the Fisher's linear discriminant method to construct the prediction model. Subsequently, Suresh et al. (2015) have proposed the RPI–Pred method, which extracts the sequence and the structural feature information of lncRNAs and proteins and the high-order 3D structural features of proteins to construct prediction models. However, the low conserved nature of lncRNA sequences makes the prediction algorithm based on lncRNA and protein feature information perform poorly in terms of accuracy and the prediction efficiency and needs to be enhanced.

(2) Heterogeneous network-based prediction model. Given the development of related experimental techniques and the accumulation of research results in the field of lncRNA, many lncRNA–protein interaction relationships have been experimentally confirmed, and researchers have successively proposed many network-based prediction algorithms to study the interaction relationships between lncRNAs and proteins. Li et al. (2015) have constructed lncRNA and protein similarity networks and combined the existing lncRNA and protein interaction data to predict unknown lncRNA–protein interaction relationships and proposed a heterogeneous network-based method called the LPIHN. The LPIHN method predicts unknown lncRNA–protein interaction relationships by constructing a heterogeneous network with the restart random walk (RWR) implemented on the constructed network to predict novel lncRNA–protein associations. Ge et al. (2016) have introduced a network dichotomy method called the LPBNI. This method performs a resource allocation procedure in the lncRNA–protein dichotomous network to evaluate candidate proteins for each lncRNA for the prediction of interaction deletions. Hu et al. (2017) have proposed a semisupervised method called the LPI–ETSLP, which reveals lncRNA–protein correlations and does not require negative samples. On the one hand, the number of known action–relationship pairs is sparse compared with the huge number of lncRNAs and proteins and directly affects the network construction and the performance of the network link prediction. On the other hand, lncRNAs or proteins with only one action–relationship in which the data behave as isolated nodes in the network and most algorithms based on network link prediction cannot effectively predict isolated nodes.

Based on the above analysis, this paper proposes a multifeature information fusion method based on lncRNA and protein sequence features and heterogeneous network topological features to predict lncRNA and protein interaction relationships. First, a novel feature extraction method based on the topological feature information of lncRNA and protein heterogeneous networks is proposed to extract the topological network features of lncRNA and protein, lncRNA sequence mutual information, the basic statistical information of lncRNA sequence bases and lncRNA expression profile features, and the evolutionary information and the composition–transition–distribution (CTD) feature information of protein sequences. Then, the above features are fused, and the fused feature information are input into the SVM to train and construct the lncRNA–protein prediction model.

MATERIALS AND METHODS

Framework of the Proposed Method

In this paper, we propose a multi-information fusion-based lncRNA–protein association prediction model consisting of three main phases, namely, (1) dataset preparation, (2) feature extraction and optimization, and (3) model training and prediction. In the dataset preparation, candidate lncRNA and protein sequences and their interaction data are usually collected from validated databases and related literature. Good training and test sets are usually required to build a high-quality prediction model. The training set is used for model training, and the test set is used to verify the transferability and the reliability of the training model. In the feature extraction and optimization, lncRNA and protein topological network features are proposed, and the protein sequence, Position Specific Scoring Matrix (PSSM), lncRNA sequence, and lncRNA expression spectrum features are extracted. Feature vectors are usually optimized by removing some irrelevant features to improve the performance of the feature information. In the model training and prediction, the SVM is used to train the input training set, and the grid search provides SVM training parameters for the construction of the training model. The prediction is performed on the given set of prediction vectors. The overall framework of the entire lncRNA–protein association prediction model is shown in **Figure 1**.

Datasets

With the development of high-throughput sequencing technologies, many public databases are available for scientists to study lncRNA–protein interactions. The NPInter database includes experimentally validated information on interactions between non-coding RNAs and other biomolecules (e.g., proteins, RNAs, and genomic DNA). The NONCODE (Liu et al., 2005) database is a comprehensive annotation database covering all types of non-coding RNAs except tRNAs and rRNAs. The NONCODE4.0 database contains 141,353 lncRNA sequence data, covering the lncRNA sequence data required in this paper. The UniProt database (Consortium, 2018) can provide the protein sequence data required in this paper. Through the abovementioned public databases, the datasets required to study lncRNA–protein interactions can be obtained and may help in the conduct of the study.

The acquisition and the preprocessing of datasets usually consist of two main steps, i.e., candidate data collection and invalid data rejection. (1) Candidate data collection, human lncRNA, and its association term data are extracted from the NPInter V2.0 database (Yuan et al., 2013; Hao et al., 2016), and 4,870 pairs of experimentally identified lncRNA–protein interaction datasets, which include 1,114 lncRNAs and 96 proteins, are obtained. Then, the lncRNA sequence information is obtained from the NONCODE 4.0 database, and the protein sequence information is obtained from the UniProt database. (2) Eliminate invalid data; since a few lncRNA sequence data are not available in some candidate datasets, proteins and lncRNAs with unavailable sequence information should be removed. In addition, some lncRNAs that only interact or are related to one protein or proteins that only interact or are related to one lncRNA have usually low correlation and potentially noisy information. Therefore, such data are excluded.

A dataset containing 4,158 lncRNA–protein interactions (including 990 lncRNAs and 27 proteins) is constructed in this paper through the above data processing steps.

Features Extraction

In this paper, five types of feature information, namely, lncRNA–protein network topology features, protein evolution information (Shao et al., 2020), protein sequence features (Liu et al., 2019), lncRNA sequence features, and lncRNA expression profile feature information, are extracted for the lncRNA–protein association prediction.

lncRNA–Protein Network Topology Features

The lncRNA–protein network can be regarded as a heterogeneous undirected graph. Suppose that the lncRNA–protein network contains N lncRNAs and M proteins and that the sets of lncRNAs and proteins are denoted by L and P, respectively, then $L = \{l_1, l_2, l_3, \ldots, l_N\}$, and $P = \{p_1, p_2, p_3, \ldots, p_M\}$. The set of edges E of this bipartite graph is denoted by $E = \{e_{ij} \mid l_i \in L,\ p_j \in P, e_{ij} = e_{ji}\}$.

If any node l_i and p_j have an interaction, then $e_{ij} = 1$, and vice versa $e_{ij} = 0$. The interaction feature L_{ij} between any lncRNA node l_i and protein node p_j is denoted as the set of edge values of node l_i and all other protein nodes except node p_j, i.e., $e_{ij} \notin L_{ij}$,$L_{ij} = \{e_{i1}, e_{ij-1}, e_{ij+1}, \ldots, e_{iM}\}$. Similarly, the interaction feature P_{ji} between any protein node p_j and protein node l_i is denoted as the set of edge values of node p_j and all other lncRNAs nodes except node l_j. Then, $e_{ji} \notin P_{ji}$,$P_{ji} = \{e_{j1}, e_{j\,i-1}, e_{j\,i+1}, \ldots, e_{jN}\}$.

The lncRNA–protein network topology is characterized as:

$$\mathrm{LPNet}_{ij} = L_{ij} \cup P_{ji},\ \ i = 1, \ldots, N, j = 1, \ldots, M. \tag{1}$$

As a result, we can obtain 1,015-dimensional network features.

Protein Evolutionary Feature Information

The protein evolutionary feature information is extracted using our previously proposed K-PSSM-composition method (Fu et al., 2018). The K-PSSM-composition feature extraction method is derived from the PSSM-composition feature extraction method. The PSSM-composition, which is proposed by Sharma et al. (2015), is used to extract protein sequence features for the prediction of the protein subcellular localization. The PSSM-composition feature extraction method can mine the evolutionary information of protein sequences but loses the mutual information between 20 amino acid residues and the local information of protein sequences. For this reason, we propose the K-PSSM-composition feature method to alleviate the above problems. In this paper, we have applied the K-PSSM-composition method to extract features from the obtained protein sequence data for the collection of the protein evolutionary feature information. The K-PSSM-composition feature is calculated as shown below.

$$\begin{aligned} &K\_PSSM\_composition \\ &= \left[PSSM\_com(1), \ldots, PSSM\_com(\lambda)\right]_{1\times(400*k)} \end{aligned} \tag{2}$$

Here, $\lambda = 1, \ldots K$; $PSSM\_com(\lambda)$ denotes the submatrix features, the calculation of which is shown in Equation (3)

$$PSSM\_\mathrm{com}(\lambda) = \left[F^A, F^R, \ldots, F^{\varphi}\right]_{1\times 400} \tag{3}$$

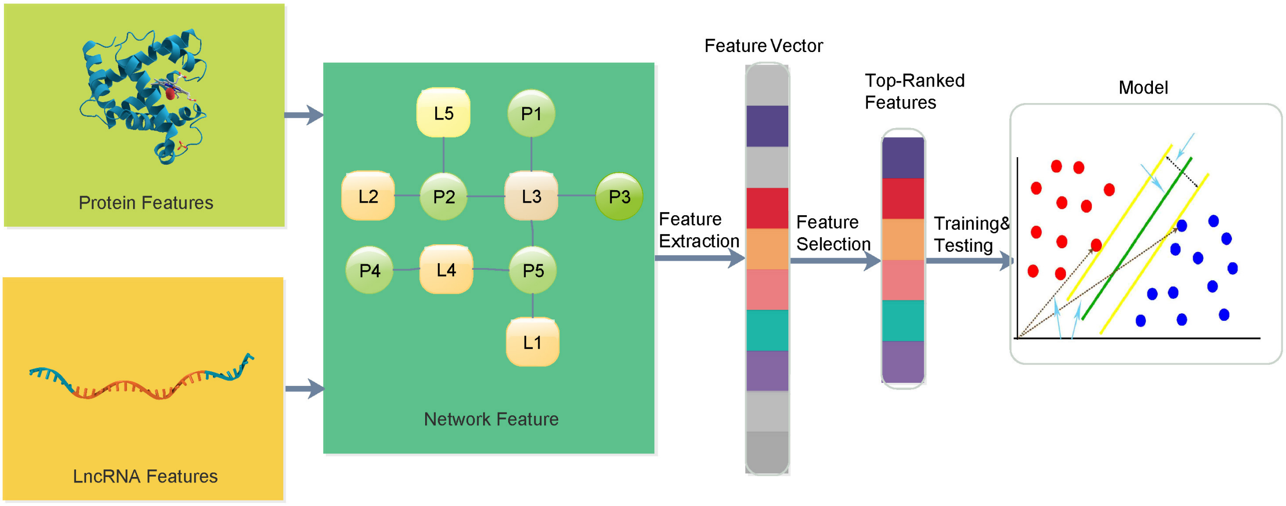


FIGURE 1 | The overall framework of the proposed method for lncRNA–protein interactions.

Here, φ denotes the 20 amino acid residues {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. F^{φ} represents the row sum of amino acid residues in the sub-PSSM matrix. In this study, $k = 1$; thus, we obtain a total of 400 dimensional features.

Protein Sequence Feature Information

In this paper, we have used the CTD (Cai et al., 2003) to extract protein sequence features, which represent the distribution patterns of specific structural or physicochemical properties in a protein or peptide sequence. Twenty amino acids are divided into three groups on the basis of different amino acid properties and represented by three feature descriptors, namely, composition (C), transition (T), and distribution (D). C denotes the percentage frequency of a specific set of amino acid properties in the calculated protein sequence, T depicts the percentage frequency of amino acids characterizing a specific property followed by another property, and D denotes the amino acid fragment describing a specific property of the whole protein sequence. Thirteen physicochemical properties have been used to calculate CTD features. Here, we use the iFeature (Chen et al., 2018) to set default parameters to extract CTD feature information and obtained a total of 504 dimensional features.

lncRNA Sequence Features

The extracted lncRNA sequence feature information contains two categories, namely, the lncRNA sequence mutual and the base compositional feature information. The lncRNA sequence mutual information is extracted using our previously proposed PSFMI feature extraction method (Fu et al., 2019) by using the entropy and the mutual information to calculate the interdependence between two bases on a given lncRNA sequence. Specifically, the 3- and the 2-gram mutual information (MI) are calculated as the characteristic information of a given lncRNA sequence.

In this study, we used entropy and MI to calculate the interdependence between bases on a given lncRNA sequence. Specifically, the 3-gram MI and the 2-gram MI were calculated separately as the characteristic information of the given lncRNA sequences. The procedure of the 3-gram triplet mutual information calculation is shown in Equation (4).

$$MI(x, y, z) = MI(x, y) - MI(x, y|z) \tag{4}$$

Here x, y, and z denote three bases that are consecutively adjacent to each other, and the equations for the calculation of MI(x, y) and conditional mutual information MI(x, $y|z$) are as follows.

$$MI(\mathrm{x, y|z}) = H(\mathrm{x|z}) - \mathrm{H(x|y, z)} \tag{5}$$

$$MI(\mathrm{x, y}) = p(\mathrm{x, y})^{*}\log(\frac{\mathrm{p(x, y)}}{\mathrm{p(x)*p(y)}}) \tag{6}$$

$$MI(\mathrm{x, y}) = MI(\mathrm{y, x}) \tag{7}$$

Where $H(x|z)$ and $H(x|y, z)$ are calculated as follows:

$$H(\mathrm{x}) = p(\mathrm{x})^{*}\log(\mathrm{p(x)}) \tag{8}$$

$$H(\mathrm{x|z}) = -\frac{p(\mathrm{x, z})}{p(\mathrm{z})}\log(\frac{p(\mathrm{x, z})}{p(\mathrm{z})}) \tag{9}$$

$$H(\mathrm{x|y, z}) = -\frac{p(\mathrm{x, y, z})}{p(\mathrm{y, z})}\log(\frac{p(\mathrm{x, y, z})}{p(\mathrm{y, z})}) \tag{10}$$

Where $p(x)$ denotes the frequency of occurrence of base x in the lncRNA sequence, $p(x, y)$ denotes the frequency of occurrence of 2 grams of bases x and y in the lncRNA sequence, and $p(x, y, z)$ denotes the frequency of occurrence of 3 grams of bases x, y, and z in the lncRNA sequence. The values of $p(x)$, $p(x, y)$, and $p(x, y, z)$ can be calculated by Equations (11)–(13) as follows.

TABLE 1 | Parameter description in the SVM-RFE + CBR method.

Parameter	Value	Describe
kerType	2	Kernel type, see libsvm. linear: 0; rbf:2
rfeC	16	Parameter C in SVM training
rfeG	0.0078	Parameter g in SVM training
useCBR	True	Whether or not use CBR
Rth	0.9	Corrcoef threshold for highly corr features

$$p(\mathrm{x}) = \frac{\mathrm{N_x} + \varepsilon}{\mathrm{L}} \tag{11}$$

$$p(\mathrm{x,y}) = \frac{\mathrm{N_{xy}} + \varepsilon}{\mathrm{L} - 1} \tag{12}$$

$$p(\mathrm{x,y,z}) = \frac{\mathrm{N_{xyz}} + \varepsilon}{\mathrm{L} - 2} \tag{13}$$

Here, N_x denotes the number of bases x that appear in the pre-miRNA sequence and L is the length of the given lncRNA sequence. The ε in Equations (11–13), denoting a very small positive real number, is used to avoid using 0 as the denominator.

For the lncRNA base composition feature information, given any lncRNA sequence, we have calculated the percentage of 4 nucleotide (i.e., A, C, G, and T) and 16 dinucleotide (e.g., AA, AG, and AC) types in each lncRNA sequence separately and obtained 20-dimensional feature vectors. The lncRNA sequence mutual information and the lncRNA base composition feature information have 19 and 16 dimensions, respectively. Thus, the total number of lncRNA sequence feature dimensions is 35; i.e., the dimensionality of the feature vector is 35 dimensions.

lncRNA Expression Profile Features

In this paper, we have obtained the lncRNA expression profile information from the NONCODE4.0 database, which contains 170,601 lncRNA expression profile data. The expression profiles describe the expression of lncRNAs in 24 types of human tissues or cells. Thus, the lncRNA expression profile features contain 24-dimensional feature vectors.

By the above analysis, we can extract a total of 1,978 (1,015 + 400 + 504 + 35 + 24) dimensional features obtained.

Feature Optimization

The feature space of lncRNA–protein interactions consists of five features, namely, lncRNA–protein network topology, lncRNA sequence, lncRNA expression profile, protein CTD information, and protein sequence evolution information features. Compared with individual features, the fusion of multiple features can capture increased sequence information, which leads to improved prediction performance. However, the fusion of multiple features produces a high-dimensional redundant feature and may lead to problems, such as excessive training time and bias in performance. Therefore, in this paper, we have used the SVM Recursive Feature Elimination (SVM-RFE) and Correlation Bias Reduction (CBR) (Yan and Zhang, 2015) to optimize the feature set.

The SVM-RFE algorithm proposed by Tolosi and Lengauer (2011) has been successfully applied to many system biology problems. The CBR algorithm has been used to reduce potential biases in linear and non-linear SVM-RFE. In this study, we use the algorithm SVM-RFE + CBR (Yan and Zhang, 2015), which consists of a combination of SVM-RFE and CBR, to optimize the feature vectors. The specific process is as follows: first, all features are ranked using SVM-RFE + CBR (Yan and Zhang, 2015) to select a set of features with the top score; second, the selected features are reorganized into new, ordered features; and finally, these new features are fed into the predictive classifier to generate a training model. Thus, we can obtain the ranked list of features through the SVM-RFE and CBR and select a set of top-ranked feature information to enable the optimal selection of features.

In the SVM-RFE + CBR method, we used the following parameters: kerType, rfeC, rfeG, useCBR, Rth. The values and descriptions of these parameters are shown in **Table 1**. The rest of the required parameters use the default settings of the SVM-RFE + CBR method.

Classification Algorithm

In this paper, we choose SVM as the classifier to build the prediction model. Specifically, the open source Library of Support Vector Machines (LIBSVM) is used for model training and construction. The LIBSVM toolbox can be downloaded for free at http://www.csie.ntu.edu.tw/~cjlin/libsvm. We integrated the toolbox in the Matrix Laboratory (MATLAB) workspace to build predictive models. The specific form of the kernel function has a large impact on the performance of the SVM. The Gaussian radial basis kernel function (RBF) has good results for non-linear classification and is widely used for bioinformatics classification; therefore, we choose RBF as the kernel function for SVM. A grid search based on five-fold cross-validation was applied to optimize the SVM parameters γ and the penalty parameter C. The grid search yielded the optimal $C = 256$ and $\gamma = 0.002$ set as their values.

Measurements

Several measures were used to evaluate the performance of the lncRNA–protein interaction prediction method comprehensively (Jin et al., 2019; Manavalan et al., 2019; Manayalan et al., 2019; Su et al., 2019a,b, 2020a,b; Qiang et al., 2020). The receiver operating characteristic curve was based on specificity and sensitivity. The area under the receiver operator characteristic curve (AUC) and the area under precision-recall curve (AUPR) were used as evaluation metrics (Wei et al., 2014, 2017a,b; Tang et al., 2020). The AUC provided a measure of classifier performance. A high AUC value indicated improved performance of the classifier. However, for class imbalance problems, the AUPR penalizes false positives in the evaluation and is more suitable than the AUC. In addition, the Matthew correlation coefficient (MCC) was used to assess the prediction performance. The MCC considered true and false and positive and negative and was usually a balanced measure that could be used even if these classes had different sizes. Sensitivity (SE), specificity (SP), precision (PR), accuracy

TABLE 2 | Performance of different feature subsets on the benchmark dataset.

Methods	ACC (%)	SE (%)	SP (%)	MCC	F1 score (%)	AUC (%)	AUPR (%)
LDNet	90.56	77.94	97.14	0.603	64.36	89.32	71.10
Pro	85.87	69.19	98.65	0.290	26.61	57.33	27.92
lRNA	84.47	52.91	**99.77**	0.067	2.83	52.29	20.34
lRNA + Pro	86.17	68.11	98.22	0.323	31.79	79.11	47.94
lRNA + LDNet	**90.81**	**78.69**	97.20	**0.615**	**65.52**	**90.99**	**73.75**
CTD + LDNet	90.62	78.25	97.18	0.606	64.62	89.02	71.32

The best values are shown in boldface.

TABLE 3 | Comparison of performance with different excellent algorithms.

Methods	ACC (%)	F1 score (%)	AUC (%)	AUPR (%)
IRWNRLPI	90.09	65.16	**91.50**	71.38
LPI–ETSLP	88.34	59.78	88.76	64.38
RWR	95.36	36.03	83.32	28.93
LPBNI	**95.81**	38.68	85.86	33.06
RPISeq–RF	46.62	14.81	39.49	6.31
RPISeq–SVM	48.23	14.93	39.87	6.98
Our method	90.82	**65.91**	90.97	**74.39**

The best values are shown in boldface.

(ACC), and MCC are defined as follows.

$$SE = \frac{TP}{TP + FN} \tag{14}$$

$$SP = \frac{TN}{TN + FP} \tag{15}$$

$$PR = \frac{TP}{TP + FP} \tag{16}$$

$$F1 - score = 2 \times \frac{SE \times PR}{SE + PR} \tag{17}$$

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{18}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TN + FP)(TP + FP)(TN + FN)}} \tag{19}$$

TP, TN, FP, and FN indicate the number of true positives, true negatives, false positives, and false negatives, respectively.

RESULTS AND DISCUSSION

Analysis of the Effect of Different Feature Information Subsets on the Experimental Performance

The effect of different feature subsets on the experimental performance was analyzed to evaluate the effect of different feature information on the lncRNA–protein prediction performance. We compared each feature subset and their two-by-two combinations on the benchmark dataset separately.

The lncRNA sequence and the lncRNA expression profile features had feature vector dimensions of 35 and 24, respectively. These features were combined for the dimensionality of the lncRNA feature information be 59 and named as lRNA features for convenience. The CTD features of protein sequences were 273 dimensions, and the K-PSSM-composition features of protein evolutionary information were 400 dimensions. The CTD and K-PSSM-composition features were combined and named as Pro features. Thus, the Pro features of proteins were 673 dimensions. The lncRNA–protein topological network features were named LDNet features, and their total feature dimension was 1,015 dimensions. Therefore, six subsets of features [i.e., lRNA, Pro, and LDNet and their two-by-two combinations (i.e., lRNA + Pro, lRNA + LDNet, and Pro + LDNet)] were obtained. To evaluate the effect and the importance of each feature subset on the prediction results, this paper uses the SVM classifier to train the prediction model, and the grid search algorithm was employed to adjust the parameters of the SVM so that each feature subset achieves the best accuracy in the same threshold range. Five-fold cross-validation tests were conducted on these six feature subsets. Experimental results are shown in **Table 2**.

The experimental results of the six feature subsets constructed in this paper by five-fold cross-validation tests are shown in **Table 2**. The ACC, SE, MCC, F score, AUC, and AUPR values of LDNet features were 90.56, 77.94, 0.603, 64.36, 89.32, and 71.10%, respectively, and higher than those of lRNA and Pro features. For the F1 score, AUC, and AUPR metrics, the LDNet features were higher by 37.75, 31.99, and 43.18%, respectively, than the Pro features, which ranked second in these three feature subsets. Therefore, the LDNet features performed the best in the separate experiments for the three feature subsets of LDNet, Pro, and lRNA, which indicated that the LDNet was the best for the lncRNA–protein association prediction because the LDNet was the largest and far exceeded the two other feature subsets.

The ACC, SE, MCC, F score, AUC, and AUPR values for lRNA + LDNet features were 90.81, 78.69, 0.615, 65.52, 90.99, and 73.75%, respectively, and were the maximum values in these six feature subsets (**Table 1**). The values of these metrics for Pro + LDNet and lRNA + LDNet feature subsets were close. The F1 score, AUC, and AUPR values for the lRNA + Pro feature subset were 31.79, 79.11, and 47.94%, respectively, which were lower than the first two combined features and even lower than the LDNet feature subset. Therefore, the lRNA + LDNet features performed best in predicting lncRNA–protein

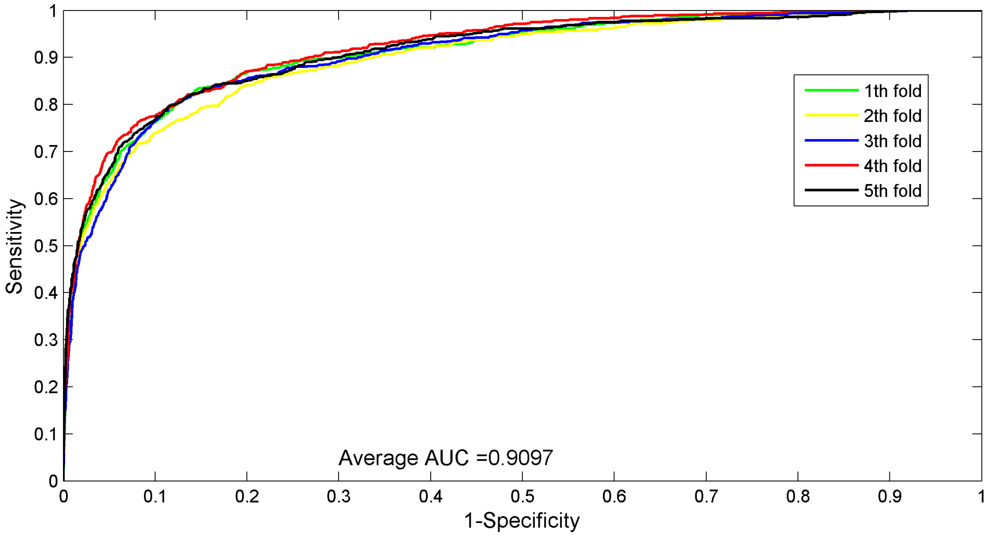


FIGURE 2 | ROC curves for five-fold cross-validation tests of the benchmark dataset.

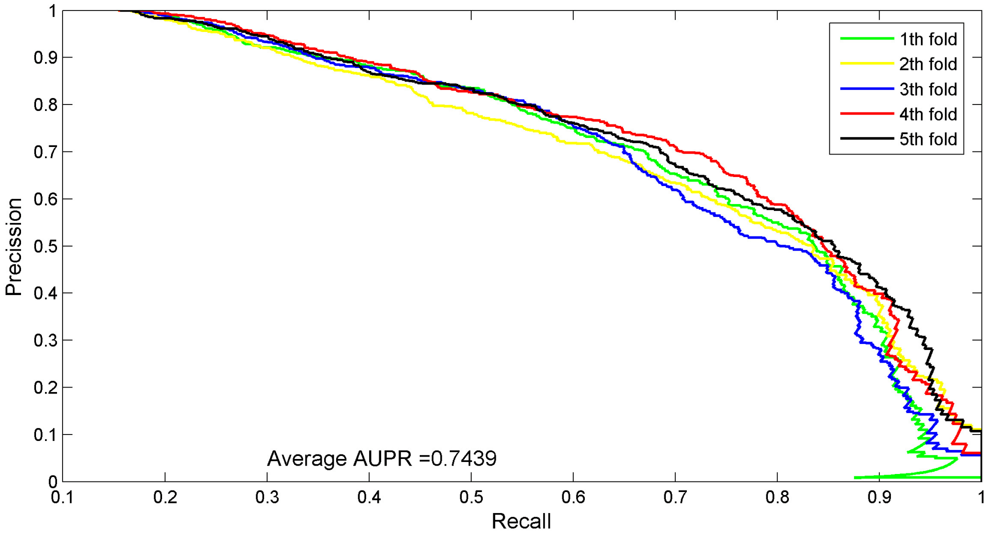


FIGURE 3 | AUPR curves for five-fold cross-validation tests of the benchmark dataset.

interactions. Among lRNA and LDNet features, the LDNet was the main decisive feature subset, which also indicated that the lncRNA and protein network topology-based features proposed in this paper had the greatest effect on the prediction performance. In addition, the performance of each feature subset in the two-by-two combination was better than the feature performance value of each feature subset individually.

Comparison With Existing Approaches

We selected the following six excellent methods for experimental comparison on the benchmark dataset to compare the performance of our proposed method with existing excellent methods. These six methods included IRWNRLPI (Zhao et al., 2018), LPI–ETSLP (Hu et al., 2017), RWR (Kohler et al., 2008), LPBNI (Li et al., 2015), RPISeq–RF (Muppirala et al., 2011), and RPISeq–SVM (Muppirala et al., 2011). The RPISeq–RF and the RPISeq–SVM models are prediction methods that extract and input lncRNA and protein features into RF or SVM predictors, whereas the IRWNRLPI, LPI–ETSLP, RWR, LPBNI, and RPISeq–RF algorithms are prediction methods that are based on heterogeneous networks constructed from lncRNAs and proteins. On the benchmark dataset, a five-fold cross-validation test was performed separately, and four evaluation metrics, namely, ACC, F1 score, AUC, and AUPR, were selected to evaluate the performance of different algorithms. Experimental results are shown in **Table 3**.

The experimental results of each evaluation index for predicting lncRNA–protein interactions are listed in **Table 3**. First, we compared the values of AUPR, which were 64.38% (LPI–ETSLP), 28.93% (RWR), 33.06% (LPBNI), 6.31% (RPISeq–RF), 6.98% (RPISeq–SVM), and 71.38% (IRWNRLPI) lower than 74.39% in our method and indicated that our method predicted reliable results.

The AUC value of our method was 90.97%, which ranked the second among all methods, and was close to the first ranked IRWNRLPI (91.50%) method and 2.21% higher than the third ranked LPI–ETSLP method. These results showed that our method had very good prediction performance. We plotted the curves of AUPR and ROC for the five-fold cross-validation tests to demonstrate the AUPR and the AUC values, respectively (**Figures 2**, **3**).

Next, we further analyzed the ACC and the F1 score values of these prediction models. The ACC of our method was 90.96% smaller than those of RWR (95.36%) and LPBNI (95.81) but better than that of IRWNRLPI (90.09%) because of very few experimentally validated lncRNA–protein interactions, which were far less than the unknown lncRNA–protein association relationships in the benchmark dataset. Therefore, the use of F1 score values to evaluate the performance of different methods than the ACC evaluation was reasonable. The F1 score value of our method was 65.91%, which was the highest among all methods and higher than those of the RWR (36.03%) and the LPBNI (38.68%). Therefore, the combined results of all experiments further demonstrated the good performance of our method in predicting lncRNA–protein associations. Notably, the four evaluation metrics (AUC, AUPR, ACC, and F1 score) of our method, which constructed prediction models on the basis of lncRNA and protein features, were more remarkable than RPISeq–RF and RPISeq–SVM.

CONCLUSIONS

lncRNAs are involved in the regulation of gene expression at the transcriptional level, epigenetics, and other life activity processes by interacting with RNA-binding proteins. Therefore, related research on the prediction of lncRNA–protein interaction relationship is beneficial in the excavation and the discovery of the mechanism of lncRNA function and action occurrence.

In this paper, a computational model for lncRNA–protein interaction relationship prediction based on the multisource information fusion is proposed. A method for representing the topological feature information of the network of lncRNA–protein interactions is proposed. Subsequently, protein evolutionary information, protein CTD sequence information features, lncRNA sequence mutual information features, and lncRNA expression profile information are extracted, and the recursive feature elimination algorithm is used to optimize feature vectors. The obtained optimized feature vectors are fed into SVM to predict lncRNA–protein interactions. Our proposed method is experimentally compared with six excellent lncRNA–protein prediction algorithms by using five-fold cross-validation tests on benchmark datasets, and experimental results show that our proposed method achieves the best performance values in AUPR and F1 score, illustrating the effectiveness and the accuracy of the proposed method in lncRNA–protein association prediction methods.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

YC, XF, and LZ conceived the concept of the work. YC, XF, and LP performed the experiments. YC, ZL, and LZ wrote the paper. All authors contributed to the article and approved the submitted version.

FUNDING

The work was supported in part by the National Natural Science Foundation of China (62002111, 61902125, and 61672223), in part by the China Postdoctoral Science Foundation (2019M662770), and in part by the Hunan Provincial Natural Science Foundation of China (2019JJ50187).

REFERENCES

Bai, Y., Dai, X., Ye, T., Zhang, P., Yan, X., Gong, X., et al. (2019). PlncRNADB: a repository of plant lncRNAs and lncRNA-RBP protein interactions. *Curr. Bioinform.* 14, 621–627. doi: 10.2174/1574893614666190131161002

Bellucci, M., Agostini, F., Masin, M., and Tartaglia, G. G. (2011). Predicting protein associations with long noncoding RNAs. *Nat. Methods* 8:444. doi: 10.1038/nmeth.1611

Cai, C. Z., Han, L. Y., Ji, Z., Chen, X., and Chen, Y. Z. (2003). SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucleic Acids Res.* 31, 3692–3697. doi: 10.1093/nar/gkg600

Cai, L., Ren, X., Fu, X., Peng, L., Gao, M., and Zeng, X. (2020a). iEnhancer-XG: Interpretable sequence-based enhancers and their strength predictor. *Bioinformatics.* btaa914. doi: 10.1093/bioinformatics/btaa914. [Epub ahead of print].

Cai, L., Wang, L., Fu, X., Xia, C., Zeng, X., and Zou, Q. (2020b). ITP-Pred: an interpretable method for predicting, therapeutic peptides with fused features low-dimension representation. *Brief. Bioinform.* bbaa367. doi: 10.1093/bib/bbaa367. [Epub ahead of print].

Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T. T., Wang, Y., et al. (2018). iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics* 34, 2499–2502. doi: 10.1093/bioinformatics/bty140

Consortium, T. U. (2018). UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.* 47, D506–D515.

Fu, X., Cai, L., Zeng, X., and Zou, Q. (2020). StackCPPred: a stacking and pairwise energy content-based prediction of cell-penetrating peptides and their uptake efficiency. *Bioinformatics* 36, 3028–3034. doi: 10.1093/bioinformatics/btaa131

Fu, X., Zhu, W., Cai, L., Liao, B., Peng, L., Chen, Y., et al. (2019). Improved pre-miRNAs identification through mutual information of pre-miRNA sequences and structures. *Front. Genet.* 10:119. doi: 10.3389/fgene.2019.00119

Fu, X., Zhu, W., Liao, B., Cai, L., Peng, L., and Yang, J. (2018). Improved DNA-binding protein identification by incorporating evolutionary information into the Chou's PseAAC. *IEEE Access* 6, 66545–66556. doi: 10.1109/ACCESS.2018.2876656

Ge, M., Li, A., and Wang, M. (2016). A bipartite network-based method for prediction of long non-coding RNA–protein interactions. *Genomics Proteomics Bioinform.* 14, 62–71. doi: 10.1016/j.gpb.2016.01.004

Hajieghrari, B., Farrokhi, N., Goliaei, B., and Kavousi, K. (2019). *In silico* identification of conserved MiRNAs from *Physcomitrella patens* ESTs and their target characterization. *Curr. Bioinform.* 14, 33–42. doi: 10.2174/1574893612666170530081523

Hao, Y., Wu, W., Li, H., Yuan, J., Luo, J., Zhao, Y., et al. (2016). NPInter v3. 0: an upgraded database of noncoding RNA-associated interactions. *Database* 2016:baw057. doi: 10.1093/database/baw057

Hong, Z., Zeng, X., Wei, L., and Liu, X. (2019). Identifying enhancer–promoter interactions with neural network based on pre-trained DNA vectors and attention mechanism. *Bioinformatics* 36, 1037–1043. doi: 10.1093/bioinformatics/btz694

Hu, H., Zhu, C., Ai, H., Zhang, L., Zhao, J., Zhao, Q., et al. (2017). LPI-ETSLP: lncRNA–protein interaction prediction using eigenvalue transformation-based semi-supervised link prediction. *Mol. Biosyst.* 13, 1781–1787. doi: 10.1039/C7MB00290D

Jeyaram, C., Philip, M., Perumal, R. C., Benny, J., Jayakumari, J. M., and Ramasamy, M. S. (2019). A computational approach to identify novel potential precursor miRNAs and their targets from hepatocellular carcinoma cells. *Curr. Bioinform.* 14, 24–32. doi: 10.2174/1574893613666180413150351

Ji, J., Tang, J., Xia, K.-,j., and Jiang, R. (2019). LncRNA in tumorigenesis microenvironment. *Curr. Bioinform.* 14, 640–641. doi: 10.2174/157489361407190917161654

Jin, Q., Meng, Z., Tuan, D. P., Chen, Q., Wei, L., and Su, R. (2019). DUNet: a deformable network for retinal vessel segmentation. *Knowl. Based Syst.* 178, 149–162. doi: 10.1016/j.knosys.2019.04.025

Kohler, S., Bauer, S., Horn, D., and Robinson, P. N. (2008). Walking the interactome for prioritization of candidate disease genes. *Am. J. Hum. Genet.* 82, 949–958. doi: 10.1016/j.ajhg.2008.02.013

Kuang, L., Zhao, H., Wang, L., Xuan, Z., and Pei, T. (2019). A novel approach based on point cut set to predict associations of diseases and LncRNAs. *Curr. Bioinform.* 14, 333–343. doi: 10.2174/1574893613666181026122045

Lambrou, G. I., Sdraka, M., and Koutsouris, D. (2019). The "Gene Cube": a novel approach to three-dimensional clustering of gene expression data. *Curr. Bioinform.* 14, 721–727. doi: 10.2174/1574893614666190116170406

Li, A., Ge, M., Zhang, Y., Peng, C., and Wang, M. (2015). Predicting long noncoding RNA and protein interactions using heterogeneous network model. *Biomed Res. Int.* 2015:671950. doi: 10.1155/2015/671950

Liu, B., Gao, X., and Zhang, H. (2019). BioSeq-Analysis2.0: an updated platform for analyzing DNA, RNA, and protein sequences at sequence level and residue level based on machine learning approaches. *Nucleic Acids Res.* 47:e127. doi: 10.1093/nar/gkz740

Liu, C., Bai, B., Skogerbø, G., Cai, L., Deng, W., Zhang, Y., et al. (2005). NONCODE: an integrated knowledge database of non-coding RNAs. *Nucleic acids research* 33(Suppl. 1), D112–D115. doi: 10.1093/nar/gki041

Lu, Q., Ren, S., Lu, M., Zhang, Y., Zhu, D., Zhang, X., et al. (2013). Computational prediction of associations between long non-coding RNAs and proteins. *BMC Genomics* 14:651. doi: 10.1186/1471-2164-14-651

Manavalan, B., Basith, S., Shin, T. H., Wei, L., and Lee, G. (2019). Meta-4mCpred: a sequence-based meta-predictor for accurate DNA 4mC site prediction using effective feature representation. *Mol. Ther. Nucleic Acids* 16, 733–744. doi: 10.1016/j.omtn.2019.04.019

Manayalan, B., Basith, S., Shin, T. H., Wei, L., and Lee, G. (2019). mAHTPred: a sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation. *Bioinformatics* 35, 2757–2765. doi: 10.1093/bioinformatics/bty1047

Mittal, N., Roy, N., Babu, M. M., and Janga, S. C. (2009). Dissecting the expression dynamics of RNA-binding proteins in posttranscriptional regulatory networks. *Proc. Natl. Acad. Sci. U.S.A.* 106, 20300–20305. doi: 10.1073/pnas.09069 40106

Munir, A., Malik, S. I., and Malik, K. A. (2019). Proteome mining for the identification of putative drug targets for human pathogen *Clostridium tetani*. *Curr. Bioinform.* 14, 532–540. doi: 10.2174/1574893613666181114095736

Muppirala, U. K., Honavar, V. G., and Dobbs, D. (2011). Predicting RNA-protein interactions using only sequence information. *BMC Bioinform.* 12:489. doi: 10.1186/1471-2105-12-489

Noureen, N., Fazal, S., Qadir, M. A., and Afzal, M. T. (2019). HCVS: pinpointing chromatin states through hierarchical clustering and visualization scheme. *Curr. Bioinform.* 14, 148–156. doi: 10.2174/1574893613666180402141107

Peng, L., Zhou, D., Liu, W., Zhou, L., Wang, L., Zhao, B., et al. (2020). Prioritizing human microbe-disease associations utilizing a node-information-based link propagation method. *IEEE Access* 8, 31341–31349. doi: 10.1109/ACCESS.2020.2972283

Qiang, X., Zhou, C., Ye, X., Du, P.-F., Su, R., and Wei, L. (2020). CPPred-FL: a sequence-based predictor for large-scale identification of cell-penetrating peptides by feature representation learning. *Brief. Bioinform.* 21, 11–23. doi: 10.1093/bib/bby091

Ray, D., Kazan, H., Cook, K. B., Weirauch, M. T., Najafabadi, H. S., Li, X., et al. (2013). A compendium of RNA-binding motifs for decoding gene regulation. *Nature* 499:172. doi: 10.1038/nature12311

Ru, X., Wang, L., Li, L., Ding, H., Ye, X., and Zou, Q. (2020). Exploration of the correlation between GPCRs and drugs based on a learning to rank algorithm. *Comput. Biol. Med.* 119:103660. doi: 10.1016/j.compbiomed.2020.103660

Shao, J., Yan, K., and Liu, B. (2020). FoldRec-C2C: protein fold recognition by combining cluster-to-cluster model and protein similarity network. *Brief. Bioinform.* bbaa144. doi: 10.1093/bib/bbaa144

Sharma, R., Dehzangi, A., Lyons, J., Paliwal, K., Tsunoda, T., and Sharma, A. (2015). Predict gram-positive and gram-negative subcellular localization via incorporating evolutionary information and physicochemical features into Chou's general PseAAC. *IEEE Trans. Nanobiosci.* 14, 915–926. doi: 10.1109/TNB.2015.2500186

Song, B., Li, K., Orellana-Martín, D., Valencia-Cabrera, L., and Pérez-Jiménez, M. J. (2020a). Cell-like P systems with evolutional symport/antiport rules and membrane creation. *Inform. Comput.* 275:104542. doi: 10.1016/j.ic.2020.104542

Song, B., Zeng, X., and Rodríguez-Patón, A. (2020b). Monodirectional tissue P systems with channel states. *Inf. Sci.* 546, 206–219. doi: 10.1016/j.ins.2020.08.030

Srivastava, N., Mishra, B. N., and Srivastava, P. (2019). *In-silico* identification of drug lead molecule against pesticide exposed-neurodevelopmental disorders through network-based computational model approach. *Curr. Bioinform.* 14, 460–467. doi: 10.2174/1574893613666181112130346

Su, R., Hu, J., Zou, Q., Manavalan, B., and Wei, L. (2020a). Empirical comparison and analysis of web-based cell-penetrating peptide prediction tools. *Brief. Bioinform.* 21, 408–420. doi: 10.1093/bib/bby124

Su, R., Liu, X., Wei, L., and Zou, Q. (2019a). Deep-Resp-Forest: a deep forest model to predict anti-cancer drug response. *Methods* 166, 91–102. doi: 10.1016/j.ymeth.2019.02.009

Su, R., Liu, X., Xiao, G., and Wei, L. (2020b). Meta-GDBP: a high-level stacked regression model to improve anticancer drug response prediction. *Brief. Bioinform.* 21, 996–1005. doi: 10.1093/bib/bbz022

Su, R., Wu, H., Xu, B., Liu, X., and Wei, L. (2019b). Developing a multi-dose computational model for drug-induced hepatotoxicity prediction based on toxicogenomics data. *Ieee Acm Trans. Comput. Biol. Bioinform.* 16, 1231–1239. doi: 10.1109/TCBB.2018.2858756

Suresh, V., Liu, L., Adjeroh, D., and Zhou, X. (2015). RPI-Pred: predicting ncRNA-protein interaction using sequence and structural information. *Nucleic Acids Res.* 43, 1370–1379. doi: 10.1093/nar/gkv020

Tang, Y.-J., Pang, Y.-H., and Liu, B. (2020). IDP-Seq2Seq: identification of intrinsically disordered regions based on sequence to sequence learning. *Bioinformaitcs* 36, 5177–5186. doi: 10.1093/bioinformatics/btaa667

Tolosi, L., and Lengauer, T. (2011). Classification with correlated features. *Bioinformatics* 27, 1986–1994. doi: 10.1093/bioinformatics/btr300

Wang, J., Wang, H., Wang, X., and Chang, H. (2020). Predicting drug-target interactions via FM-DNN learning. *Curr. Bioinform.* 15, 68–76. doi: 10.2174/1574893614666190227160538

Wang, L., Xuan, Z., Zhou, S., Kuang, L., and Pei, T. (2019). A novel model for predicting LncRNA-disease associations based on the LncRNA-MiRNA-disease interactive network. *Curr. Bioinform.* 14, 269–278. doi: 10.2174/1574893613666180703105258

Wang, Y., Chen, X., Liu, Z.-P., Huang, Q., Wang, Y., Xu, D., et al. (2013). *De novo* prediction of RNA–protein interactions from sequence information. *Mol. Biosyst.* 9, 133–142. doi: 10.1039/C2MB25292A

Wei, L., Liao, M., Gao, Y., Ji, R., He, Z., and Zou, Q. (2014). Improved and promising identification of human MicroRNAs by incorporating a high-quality negative set. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 11, 192–201. doi: 10.1109/TCBB.2013.146

Wei, L., Wan, S., Guo, J., and Wong, K. K. L. (2017a). A novel hierarchical selective ensemble classifier with bioinformatics application. *Artif. Intell. Med.* 83, 82–90. doi: 10.1016/j.artmed.2017.02.005

Wei, L., Xing, P., Zeng, J., Chen, J., Su, R., and Guo, F. (2017b). Improved prediction of protein-protein interactions using novel negative samples, features, and an ensemble classifier. *Artif. Intell. Med.* 83, 67–74. doi: 10.1016/j.artmed.2017.03.001

Xiao, Q., Luo, J., and Dai, J. (2019). Computational prediction of human disease-associated circRNAs based on manifold regularization learning framework. *IEEE J. Biomed. Health Inform*. 23, 2661–2669. doi: 10.1109/JBHI.2019.2891779

Xiao, Q., Luo, J., Liang, C., Cai, J., and Ding, P. (2018). A graph regularized non-negative matrix factorization method for identifying microRNA-disease associations. *Bioinformatics* 34, 239–248. doi: 10.1093/bioinformatics/btx545

Xiao, Q., Zhang, N., Luo, J., Dai, J., and Tang, X. (2020). Adaptive multi-source multi-view latent feature learning for inferring potential disease-associated miRNAs. *Brief. Bioinform*. bbaa028. doi: 10.1093/bib/bbaa028. [Epub ahead of print].

Yan, K., and Zhang, D. (2015). Feature selection and analysis on correlated gas sensor data with recursive feature elimination. *Sensors Actuat. B Chem.* 212, 353–363. doi: 10.1016/j.snb.2015.02.025

Yuan, J., Wu, W., Xie, C., Zhao, G., Zhao, Y., and Chen, R. (2013). NPInter v2.0: an updated database of ncRNA interactions. *Nucleic Acids Res.* 42, D104–D108. doi: 10.1093/nar/gkt1057

Zeng, X., Liao, Y., Liu, Y., and Zou, Q. (2017a). Prediction and validation of disease genes using HeteSim scores. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 14, 687–695. doi: 10.1109/TCBB.2016.2520947

Zeng, X., Lin, W., Guo, M., and Zou, Q. (2017b). A comprehensive overview and evaluation of circular RNA detection tools. *PLoS Comput. Biol.* 13:e1005420. doi: 10.1371/journal.pcbi.1005420

Zeng, X., Liu, L., Lü, L., and Zou, Q. (2018). Prediction of potential disease-associated microRNAs using structural perturbation method. *Bioinformatics* 34, 2425–2432. doi: 10.1093/bioinformatics/bty112

Zeng, X., Zhu, S., Liu, X., Zhou, Y., Nussinov, R., and Cheng, F. (2019). deepDR: a network-based deep learning approach to *in silico* drug repositioning. *Bioinformatics* 35, 5191–5198. doi: 10.1093/bioinformatics/btz418

Zeng, X., Zhu, S., Lu, W., Liu, Z., Huang, J., Zhou, Y., et al. (2020). Target identification among known drugs by deep learning from heterogeneous networks. *Chem. Sci.* 11, 1775–1797. doi: 10.1039/C9SC04336E

Zhang, L., He, Y., Wang, H., Liu, H., Huang, Y., Wang, X., et al. (2019). Clustering count-based RNA methylation data using a nonparametric generative model. *Curr. Bioinform.* 14, 11–23. doi: 10.2174/1574893613666180601080008

Zhang, X., Zou, Q., Rodriguez-Paton, A., and Zeng, X. (2019). Meta-path methods for prioritizing candidate disease miRNAs. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 16, 283–291. doi: 10.1109/TCBB.2017.2776280

Zhang, Y., and Zou, Q. (2020). PPTPP: A novel therapeutic peptide prediction method using physicochemical property encoding and adaptive feature representation learning. *Bioinformatics* 36, 3982–3987. doi: 10.1093/bioinformatics/btaa275

Zhao, Q., Zhang, Y., Hu, H., Ren, G., Zhang, W., and Liu, H. (2018). IRWNRLPI: integrating random walk and neighborhood regularized logistic matrix factorization for lncRNA-protein interaction prediction. *Front. Genet.* 9:239. doi: 10.3389/fgene.2018.00239

Zou, Q., Lin, G., Jiang, X., Liu, X., and Zeng, X. (2020). Sequence clustering in bioinformatics: an empirical study. *Brief. Bioinform.* 21, 1–10. doi: 10.1093/bib/bby090

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers
in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 14 May 2021
doi: 10.3389/fbioe.2021.674211

Pepblock Builder VR – An Open-Source Tool for Gaming-Based Bio-Edutainment in Interactive Protein Design

Venkata V. B. Yallapragada[1,2*†], Tianshu Xu[3†], Sidney P. Walker[1,2], Sabin Tabirca[3,4‡] and Mark Tangney[1,2,5,6*‡]

[1] Cancer Research @ UCC, University College Cork, Cork, Ireland, [2] SynBioCentre, University College Cork, Cork, Ireland, [3] School of Computer Science and Information Technology, University College Cork, Cork, Ireland, [4] Department of Computer Science, Transylvania University of Braşov, Braşov, Romania, [5] APC Microbiome Ireland, University College Cork, Cork, Ireland, [6] iEd Hub, University College Cork, Cork, Ireland

OPEN ACCESS

Edited by:
Zhibin Lv,
University of Electronic Science and Technology of China, China

Reviewed by:
Lijun Dou,
Shenzhen Polytechnic, China
Mouzhe Xie,
University of Chicago, United States

****Correspondence:***
Venkata V. B. Yallapragada
vvb.yallapragada@cit.ie
Mark Tangney
m.tangney@ucc.ie

†These authors have contributed equally to this work and share first authorship

‡These authors share last authorship

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *28 February 2021*
Accepted: *24 March 2021*
Published: *14 May 2021*

Citation:
Yallapragada VVB, Xu T, Walker SP, Tabirca S and Tangney M (2021) Pepblock Builder VR – An Open-Source Tool for Gaming-Based Bio-Edutainment in Interactive Protein Design. Front. Bioeng. Biotechnol. 9:674211. doi: 10.3389/fbioe.2021.674211

Proteins mediate and perform various fundamental functions of life. This versatility of protein function is an attribute of its 3D structure. In recent years, our understanding of protein 3D structure has been complemented with advances in computational and mathematical tools for protein modelling and protein design. 3D molecular visualisation is an essential part in every protein design and protein modelling workflow. Over the years, stand-alone and web-based molecular visualisation tools have been used to emulate three-dimensional view on computers. The advent of virtual reality provided the scope for immersive control of molecular visualisation. While these technologies have significantly improved our insights into protein modelling, designing new proteins with a defined function remains a complicated process. Current tools to design proteins lack user-interactivity and demand high computational skills. In this work, we present the Pepblock Builder VR, a gaming-based molecular visualisation tool for bio-edutainment and understanding protein design. Simulating the concepts of protein design and incorporating gaming principles into molecular visualisation promotes effective game-based learning. Unlike traditional sequence-based protein design and fragment-based stitching, the Pepblock Builder VR provides a building block style environment for complex structure building. This provides users a unique visual structure building experience. Furthermore, the inclusion of virtual reality to the Pepblock Builder VR brings immersive learning and provides users with "being there" experience in protein visualisation. The Pepblock Builder VR works both as a stand-alone and VR-based application, and with a gamified user interface, the Pepblock Builder VR aims to expand the horizons of scientific data generation to the masses.

Keywords: virtual reality, protein gaming, molecular visualisation, edutainment, 3D structure

INTRODUCTION

Proteins mediate and perform various fundamental functions of life. This versatility of protein function is an attribute of its 3D structure. Understanding the protein 3D structure is crucial for various fields of science. Elucidating the structure of a protein revolutionised the field of protein science and paved the way for the establishment of massive databases. Traditionally, physical methods such as NMR spectroscopy and X-ray crystallography have been deployed to elucidate and study the 3D structure of proteins. The recent advances in computational sciences have resulted in sophisticated algorithms for predicting and modelling the 3D structure of a protein from its corresponding amino acid sequence. Designing entirely novel proteins with a defined structure is also feasible with the current developments in *de novo* protein design (Huang et al., 2016). Traditionally, proteins were primarily designed by sequence-based design. In recent years, other concepts such as parametric modelling (Wood et al., 2017) and fragment assembly (Huang et al., 2011) have been developed to effectively design protein structures with particular functions. In spite of the recent progress, the field of protein design has a steep learning curve and requires great technical skills to effectively design novel protein structures. This lack of a complete visual-based interactive design tool is one of the bottlenecks in translating the protein design technology toward a broader creative audience. Recently (Yeh et al., 2018), Yeh et al. introduced a building block style graphical interface Elfin UI, which provides a modular approach to structure building. Such visual-based interactive interfaces create a platform for users with limited technical knowledge to create novel protein structures with minimal/no focus on the amino acid sequence.

Visualising Proteins in 3D and Why?

Visualising proteins in 3D has various unique applications and advantages for both scientific and educational purposes (**Figure 1**). In education, teaching the protein structure is a highly complicated task (Richardson and Richardson, 2002). The differences between the primary, secondary, and tertiary structures of proteins are challenging to understand. The non-linearity in protein folding, stereo-configurations and formation of large complex assemblies add to the existing hurdles in teaching the protein structure. In such cases, deploying computational 3D visualisation tools to visualise proteins not only provides users with an enhanced visual experience but also increases interest in the scientific field (Cai et al., 2021).

From a scientific perspective, visualisation of the 3D structure of a protein forms a pivotal component in both modelling the protein structure (physical methods and/or computational prediction) and *de novo* protein design. (i) *For biochemists:* It provides insights into various protein domains such as hydrophobic regions, active sites, catalytic sites etc. (ii) *For evolutionary biologists:* Visualising and mapping 3D structures aids in studying homology in structure. (iii) *In drug designing:* Visualising protein–protein interactions and protein interactions with small molecules can be key to understanding drug activity. (iv) *In de novo protein design:* Visualising the designed backbone structures and superimposing the designed structures with experimental structures is very commonly performed using *in silico* molecular visualisation.

Current Tools for Visualising Proteins

Over the years, a wide variety of visualisation tools have been developed and deployed for 3D structure visualisation. These tools range from printout stereoscopic images to sophisticated VR CAVEs. Early 21st century saw a sudden increase in the number of protein structures deposited in protein databases (Bank, 2021), and a need for better visualisation tools increased in parallel. PyMOL (2021), VMD (Humphrey et al., 1996), Chimaera (Pettersen et al., 2004) and Rasmol (Sayle and Milner-White, 1995) are some examples of widely used stand-alone applications for molecular visualisation. Later, web-based applications such as Jmol (2021) and iView (Li et al., 2014) have gained interest in the scientific community. The web-based applications provided a way to integrate the visualisation into websites. Recently, several mobile-based applications for Android and iOS have also been developed by various groups for molecular visualisation. Although these tools provide a wide canvas of features for visualising proteins in 3D, there are a few considerable drawbacks such as the lack of full visual immersion and the lack of a real 3D effect.

Virtual reality (VR) based methods provide an alternative solution for these problems (Goddard et al., 2018a). Since its invention in the late 1960s, a wide range of technologies have been researched and deployed for VR (Indhumathi et al., 2007). Head mount devices (HMDs) such as Oculus Rift, Cave automatic virtual environment (CAVEs) and smartphone-based Google cardboard-like devices are some well-known examples. The introduction of VR brings a full immersive visual experience to molecular visualisation. The 3D effect generated in virtual reality provides other advantages such as enhanced accuracy in colour depiction of the models, accurate depth perception, improved wide field of view, haptic interfacing and better molecular viewing resolution (Norrby et al., 2015).

Visualising biomolecules (proteins in particular) in VR has gained wide attention recently (Goddard et al., 2018a). Tools such as ChimeraX (Goddard et al., 2018b), BioVR (Zhang et al., 2019), and StarCave (DeFanti et al., 2009) (cave based) have been developed for visualising the 3D structure of proteins in VR. Although the current technology provides a plethora of functionalities for the user, the potential of molecular visualisation in VR is still a maturing field. Easier navigation in the VR environment, better user interface (UI), faster rendering and simplified instrumentation are some areas that are expected to see some improvements in the near future. Parallel advancements in affordable VR headsets and increasing computational power and graphics project interesting times ahead.

Gamification in Education and Scientific Research

Computer games are powerful audio-visual teaching tools and have been used as interactive learning aids in various fields of education. Gamification of scientific learning is becoming a popular form of edutainment. Today, edutainment is a powerful form of experiential smart learning (Anikina and Yakimenko, 2015), and the market value of edutainment is projected to reach 11.34 billion by 2028 (Global Edutainment Market Growth, 2021). Through fun-based learning, gamification instils curiosity, motivated experience and interest in learning (Cai et al., 2006, 2008). With the advances in human–computer interaction strategies, gamification has also expanded the realms of research

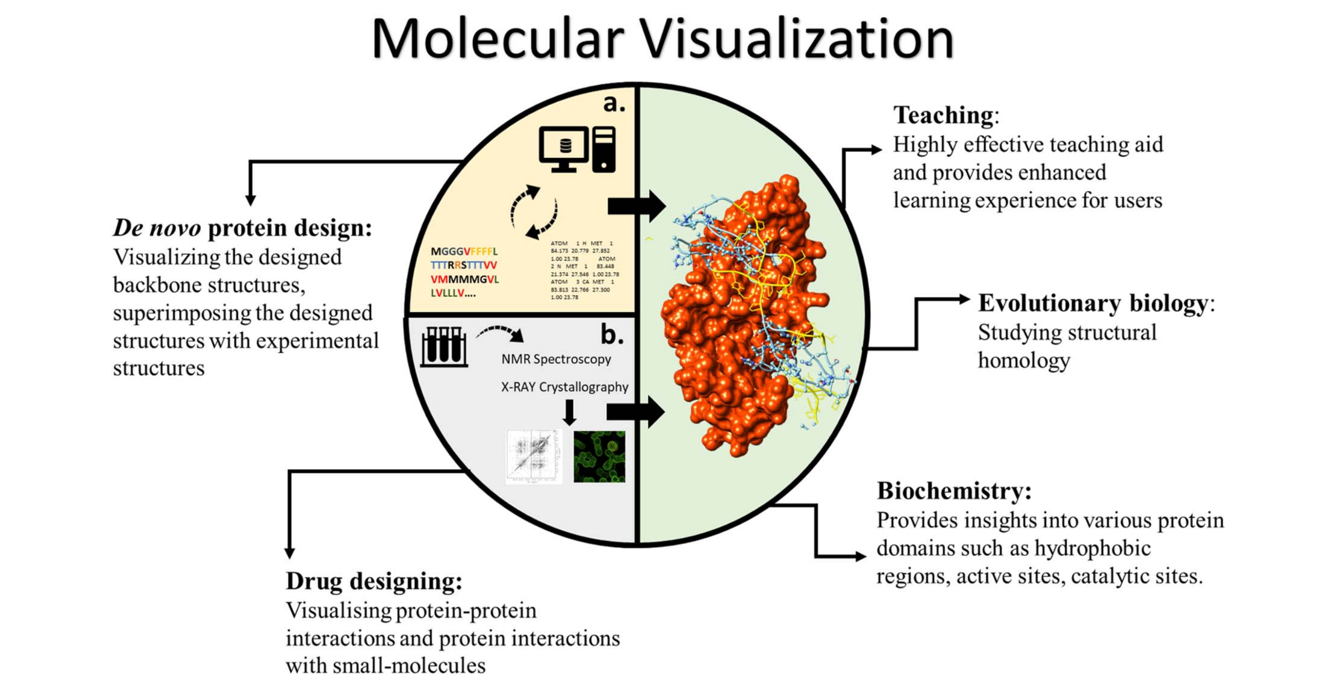
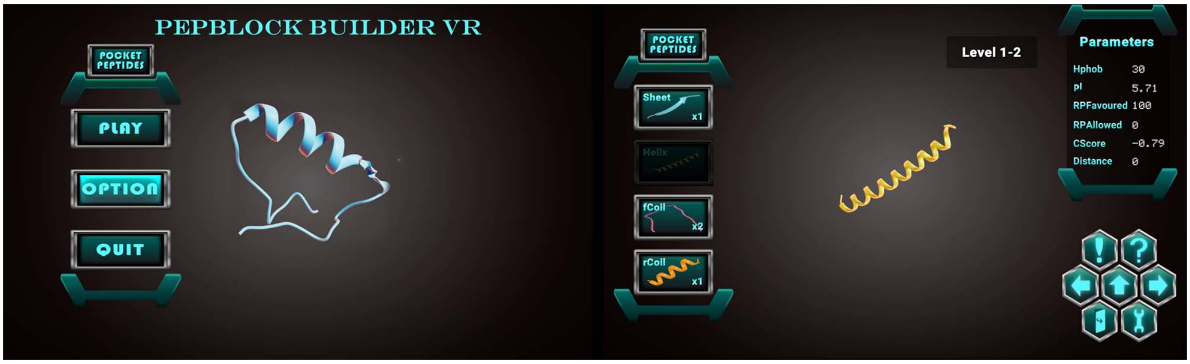

Molecular Visualization
a.
MGGGVFFFFL
TTTRRSTTTVV
VMMMMGVL
LVLLLV....
ATOM 1 H MET 1
84.173 20.779 27.852
1.00 23.78 ATOM
2 N MET 1 83.448
21.374 27.546 1.00 23.78
ATOM 3 CA MET 1
83.813 22.766 27.300
1.00 23.78
b.
NMR Spectroscopy
X-RAY Crystallography
De novo protein design: Visualizing the designed backbone structures, superimposing the designed structures with experimental structures
Teaching: Highly effective teaching aid and provides enhanced learning experience for users
Evolutionary biology: Studying structural homology
Biochemistry: Provides insights into various protein domains such as hydrophobic regions, active sites, catalytic sites.
Drug designing: Visualising protein-protein interactions and protein interactions with small-molecules

FIGURE 1 | Various applications of molecular visualisation. The interior graphics refer to **(A)** connection between computational "protein modelling and design" and visualisation. **(B)** Showing the connection between physical structure elucidation methods and visualisation.

TABLE 1 | Examples of popular games for science and bio-edutainment.

Game/Tool	Scientific problem addressed	Description
Foldit (Kleffner et al., 2017)	Protein folding	Game involving real-time manipulation of protein structures, with results used to solve real-life problems. Recently resolved the structure of HIV-associated enzyme.
EteRNA (Anderson-Lee et al., 2016)	RNA folding	Puzzle-based game to provide insights into RNA design.
The Cure (Good et al., 2014)	Phenotyping in breast cancer	Detection of molecular signatures linked to specific breast cancer prognosis through web-based game.
Phylo (Kwak et al., 2013)	DNA sequence alignment	Web-based Tetris style game facilitating sequence alignments.
EyeWire (Cooper et al., 2018)	Neuronal mapping	Web-based 3D neuron reconstruction game.
Brainflight (Brainflight, 2021)	Neuronal mapping	Game to track the path of electric impulses travelling through the brain.



Parameters

Hphob	30
pI	5.71
RPFavoured	100
RPAllowed	0
CScore	-0.79
Distance	0

FIGURE 2 | (A) Welcome screen of the Pepblock Builder VR showing the menu panel for play, options (sounds, music, and volume controls) and quit buttons **(B)** The game UI of the Pepblock Builder VR with the peptide panel on the left with basic secondary structures as building blocks and the right panel showing various *in silico* parameters related to the structure. The hexagonal control panel contains functions for play instructions (! symbol), undo and redo (right and left arrows), help button (! symbol), submit button (up arrow), exit button (with door symbol) and a settings toggle button.

by taking advantage of the massive number of citizen scientists to contribute toward complicated scientific goals. When combined with audio-visual edutainment and gamification, scientific goals appeal to a larger audience. The path from concept building, world realisation and problem solving are an integral part of gaming. This path is commonly observed in real-life scenarios such as scientific research. By mimicking scientific research and by adding gamification principles, tools/games for bio-edutainment such as fold.it and Phylo have gained public attention and resulted in remarkable scientific achievements (Koepnick et al., 2019) (**Table 1**).

Introducing Pepblock Builder VR

The Pepblock Builder VR is a gamified protein visualisation and design tool for bio-edutainment and fully visual-based protein structure building. The Pepblock Builder VR is currently available in two versions: (i) a stand-alone desktop-based tool and (ii) an HMD-based tool in the VR environment. The Pepblock Builder VR is a versatile tool that can serve as a visualisation tool, an edutainment tool and a structure-based modular design tool. As a visualisation tool, the Pepblock Builder provides an immersive experience to the users to visualise complicated 3D structures. As an edutainment tool, the Pepblock Builder VR provides an interactive and experiential learning experience to users through gamification. As a design tool, the modular (semi-LEGO® style) protein design approach of the Pepblock Builder VR offers a multitude of non-technical users to build complex protein structures.

MATERIALS AND METHODS

Ethics

Written informed consent was obtained from the [individual(s) and/or minor(s)' legal guardian/next of kin] for the publication of any potentially identifiable images or data included in this article.

PC Infrastructure Used for Pepblock Builder VR

The Pepblock Builder VR was developed on a desktop PC running Windows 10, with 8 GB RAM, 4GB NVidia graphics card and Intel core I7 processor (7th generation). Two 16-inch LCD monitor screens were used for display. Standard keyboard and optical mouse were used as input devices. The development of the Pepblock Builder VR demanded a high-spec configuration to facilitate multitasking during the design, implementation and testing stages of various iterations of the software in parallel (**Table 2**).

VR Setup

Oculus Rift and Oculus Rift S setups were used for the development of the Pepblock Builder VR for virtual reality environment. The setup included an HMD, two handheld controllers for human–VR interaction and two stand sensors. The program has a pre-built library called OVR Utilities Plugin (Oculus Integration) for implementing functionalities with Oculus Rift. This enables the connection of external parts such as controllers, headset etc., to the VR environment.

TABLE 2 | Key features of Pepblock Builder VR.

Technical Features	Details/Format
File types	PDB, X3D
Structure visualisation format	Ribbons and Cartoons
VR module	Oculus Rift
Human interaction in VR and PC	Handheld controllers and optical mouse
Manipulation features	Rotation, zooming, bending and twisting carbon alpha chains, 360 X, Y, Z movements
GUI model and scheme	Space neon colour scheme
Graphics and Game mechanics tools used	Blender and Unity game engine

Data Generation, Protein Modelling and *in silico* Parameters

Protein tertiary structures were generated by the I-TASSER suite (v5.1). C-scores for all the structures were obtained from I-Tasser. *In silico* parameters such as hydrophobicity and theoretical pI were calculated using the ProtParam facility, hosted by Expasy. Saves server (using the Verify3D utility) was used to generate the Ramachandran plots and scores for all the structures modelled using I-Tasser. RMSD scores between the atoms of the guided shadow and the query protein are calculated through in-house scripts written in Python and R, developed at the Tangney lab. Amino acid sequences and/or PDB files were given as inputs wherever necessary.

Programming

Scripts for processing the *in silico* parameter data from the servers and integrating into the Unity environment were written in Python and C# programming language. Individual scripts used in integrating the data from various servers and scripts used for calculating certain *in silico* parameters such as the RMSD score are documented in GitHub[1].

Graphics and Game Mechanics

Blender v2.8 was used (i) for generating all the graphical 3D files using protein structures (modelled using I-Tasser), (ii) for guided shadows, and (iii) to add flexibility to protein backbones. The Unity v2019.2.9 personal edition game engine was used for animating the opening cutscene, internal game mechanics and implementation.

IMPLEMENTATION AND RESULTS

Pepblock Builder VR as an Edutainment Tool

The Unity 3D game engine was used to develop the Pepblock Builder VR's gameplay and graphics. Two versions of the

[1] https://github.com/TIanshuXu/Pocket-Peptides-PC

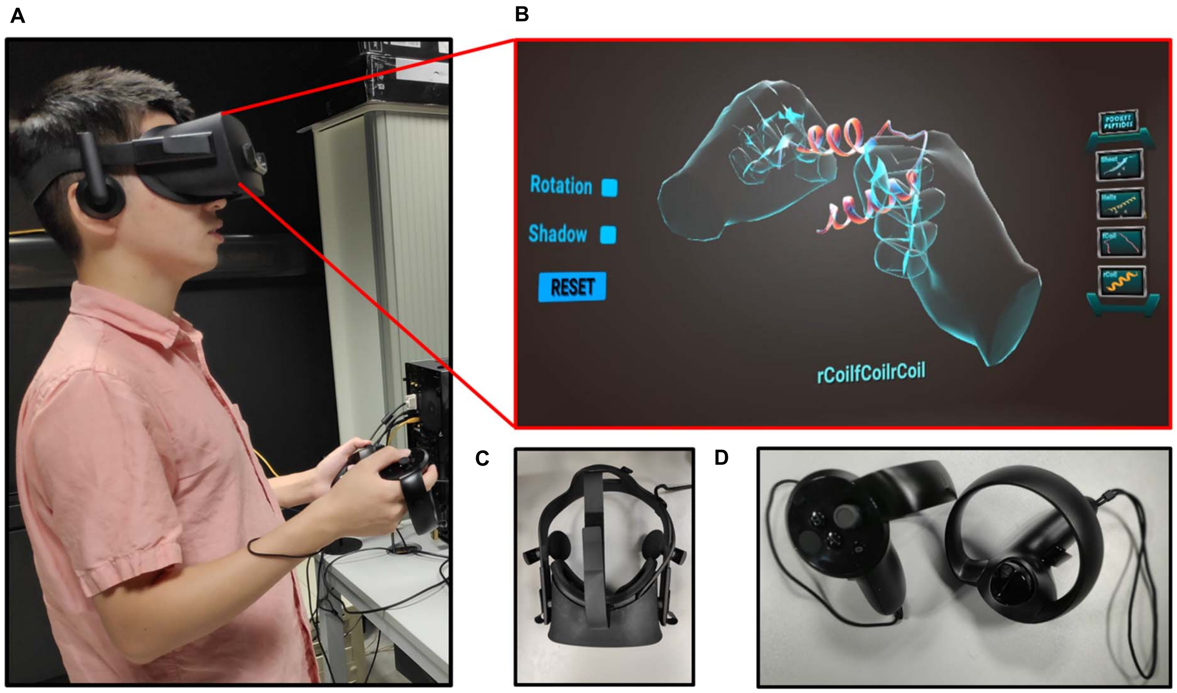


FIGURE 3 | (A) User with Oculus Rift headset and controllers, experiencing the Pepblock Builder VR in VR. **(B)** UI of the Pepblock Builder VR VR. **(C)** Oculus rift headset used for the Pepblock Builder VR. **(D)** Handheld controllers used for user–computer interaction.

Pepblock Builder VR were developed, i.e., (i) Pepblock Builder VR desktop version and (ii) Pepblock Builder VR VR version. **Figure 2** shows the welcome screen and game UI for the desktop version. **Figure 3** shows the VR UI and Oculus Rift S VR setup. Neon blue and Neon green colour scheme was used throughout the UI to give a space Sci-Fi effect.

Storyboard and Game Narrative

Modern gaming benefits from the inclusion of a captivating storyboard to narrate the game world and to introduce in-game rules to the players. The Pepblock Builder VR has a passive narrative of a human-destroyed earth and the grand ecosystem that was lost due to human activities. An AI bot seeks help for building new protein structures using three fundamental secondary shapes, i.e., Helix, Coil, and Sheet. The 60-s opening cutscene shows an animated post-apocalyptic tutorial and guides the user into building new proteins to restore life on earth. Between each task and level, an in-game screensaver presents facts about proteins displayed in curated graphics to deepen the user's understanding of proteins. **Figure 4** shows screenshots from the opening cutscene of the Pepblock Builder VR.

Interface, Gameplay and Functionalities

The Pepblock Builder VR gameplay is based on simple LEGO style modular protein building. Users are engaged in creating novel structures based on the challenges provided by each level. The Pepblock Builder VR is designed with a progressive level-up approach to guide, teach and challenge the users toward complicated design problems using proteins.

Initially (see **Figure 5**), the users are provided with a blank canvas in the centre, where any structure from the left panel could be dragged and dropped. The left panel has four basic shapes. The panel on the right displays the *in silico* parameters of the current structure in display. The control panel at the bottom right has a help button to provide a tour of the interface and a challenge button to display the "challenge" of the current level. The left and right arrows provide undo and redo options, respectively, while the centre up button submits the user response.

Game users are required to construct the given shape (neon blue shadow) using the basic shapes from the left panel. Subsequent levels include the ability to modify the structure by rotating, twisting and bending the protein backbone. A haptic-snaplock feature locks the protein into the shadow when the complete resemblance is achieved. For the PC version, users are allowed to click on buttons in the UI to either navigate through the various game scenes (menus and different levels) or toggle protein components (secondary structure) and tips. The protein components (displayed as icons on the left in the game scenes) are available to be dragged and dropped on the glowing area (the centre of the screen). The users are also allowed to rotate and zoom the whole scene by clicking the middle mouse button. In advanced levels, users also unlock bending, twisting and turning features by clicking and dragging on any point on the 3D structure.

The progressive level approach of the Pepblock Builder VR increases the complexity of the structures and challenges with increasing difficulties. The values in the *in silico* parameter panel, relevant to the structure displayed, change as the user modifies

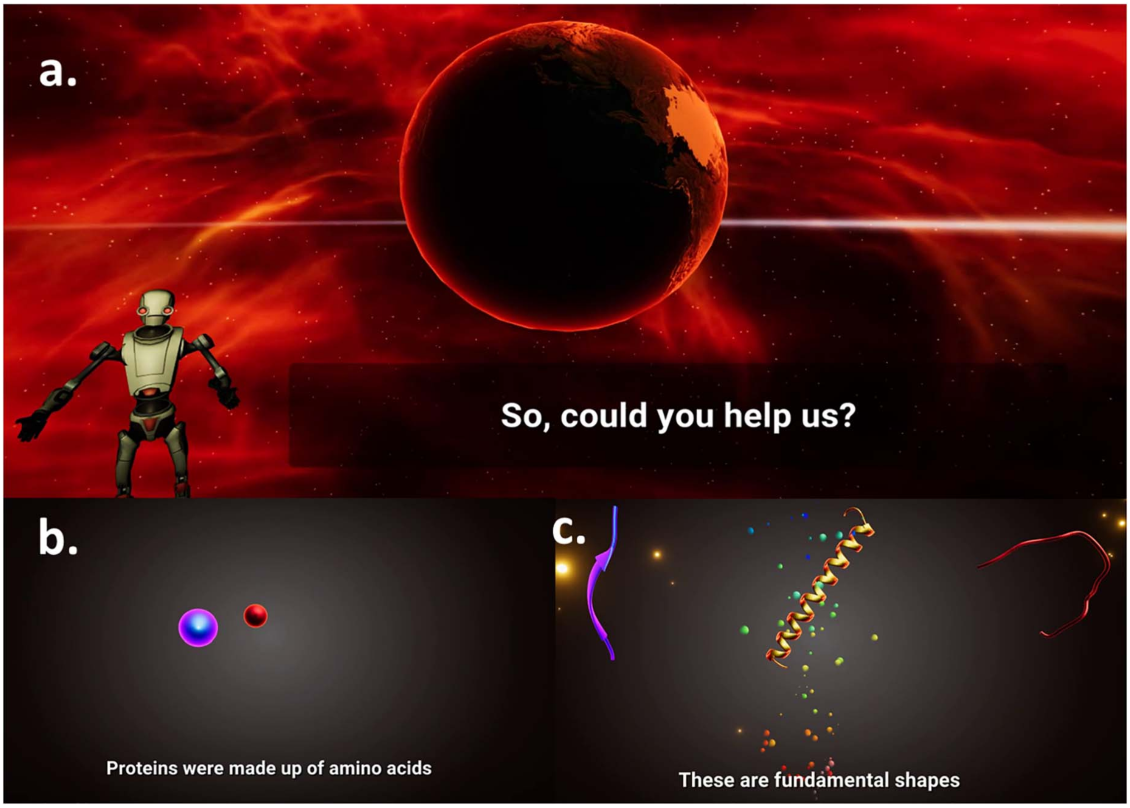

a.
So, could you help us?
b.
Proteins were made up of amino acids
c.
These are fundamental shapes

FIGURE 4 | Screenshots from storyboard of the Pepblock Builder VR (Cutscene). The Pepblock Builder VR is set up in a post-apocalyptic world. **(a)** An AI guides a tutorial, talking about the grand ecosystem that existed and is now seeking help to rebuild the world. **(b)** Concepts on the composition of proteins are explained in an animated fashion. **(c)** Secondary structures of proteins (Helices, Coils, and Sheets) are projected as fundamental building blocks (LEGO style).

the structure. This promotes the basic understanding of the relationship between the *in silico* parameters and the structure of a protein. There is potential to implement more advanced levels where the challenges would demand direct structure design for a defined set of *in silico* parameters.

Pepblock Builder VR in Virtual Reality

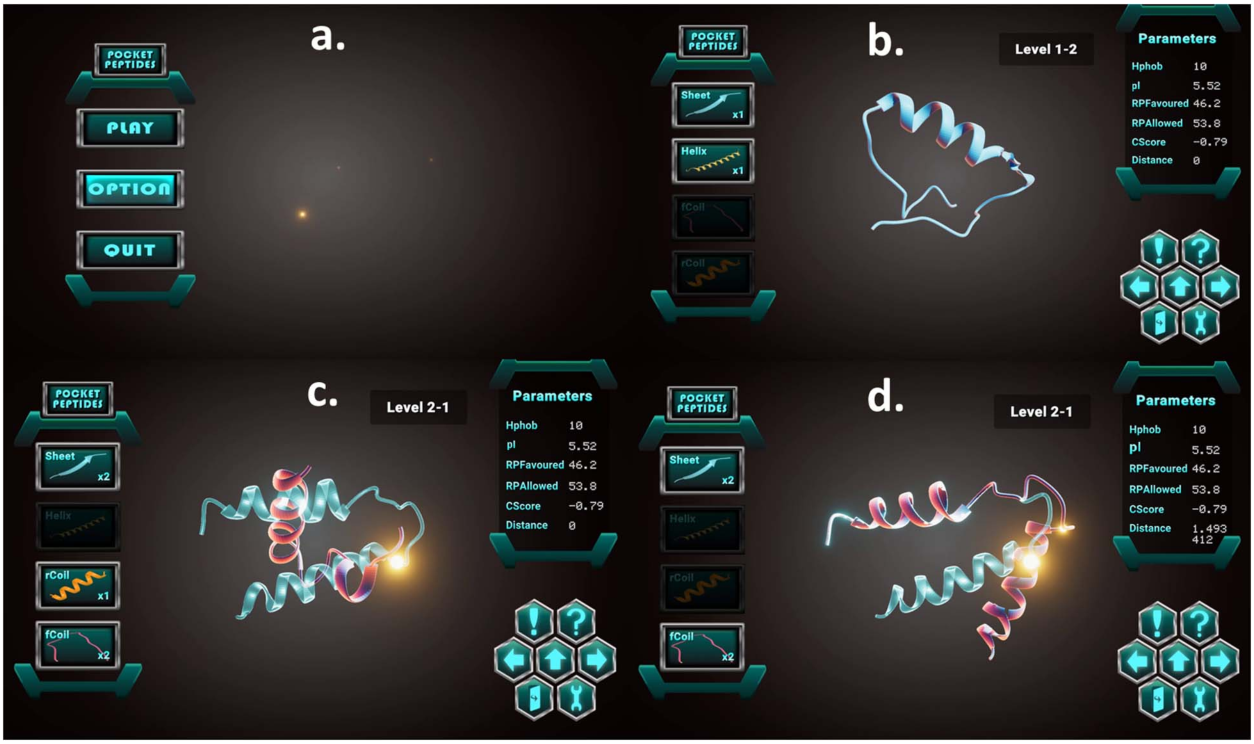
Although the game storyboard, narrative and game goals remain the same in VR as were in the desktop game, the user interface and user interaction were reprogrammed. The "drag and drop" feature was changed to grab and throw, which is a common user interaction method in various VR games. Controller buttons and actions for all the in-game user interactions are listed in **Box 1**. Users are allowed to walk around in the VR environment (available to Oculus Rift or higher). Similar to the stand-alone version, users can click on protein component icons on their left-hand side by clicking on the right trigger button on their Touch controller to display the corresponding protein model. Once a protein model is displayed at the centre of the scene, the player can grab protein models behind the protein component icons by tapping and holding the grip button on a Touch controller. After grabbing a protein model, they can examine it and be able to throw it onto the large displaying model at the centre of the scene. When a second protein model is thrown upon the existing structure, the resultant (modelled) combination of the two complexes is displayed.

Pepblock Builder VR as a Visualisation Tool

The gaming principles and the easy-to-navigate UI are the two key features of the Pepblock Builder VR. While the gamification forms the foundation for bio-edutainment and understanding modular protein design, the UI of the Pepblock Builder VR could be deployed to visualise any protein of interest. Any file in the.PDB format can be converted and automatically processed to be visualised in the in-game virtual environment. In the current version (the automatic processing mode), the proteins are depicted in ribbons and cartoon format by default. Other depiction models can also be achieved by manually converting any.PDB file to a.DAE file and by importing into the Pepblock Builder VR UI. An example of visualising a protein downloaded from RCSB PDB databases is shown in **Figure 6**.

Pepblock Builder VR as a Protein Design Tool

The Pepblock Builder VR provides the user with a panel of LEGO-style protein building blocks. Both in the stand-alone and



Parameters
Hphob 10
pI 5.52
RPFavoured 46.2
RPAllowed 53.8
CScore -0.79
Distance 0

Parameters
Hphob 10
pI 5.52
RPFavoured 46.2
RPAllowed 53.8
CScore -0.79
Distance 0

Parameters
Hphob 10
pI 5.52
RPFavoured 46.2
RPAllowed 53.8
CScore -0.79
Distance 1.493 412

FIGURE 5 | Screenshots showing the game interface and toggle buttons: **(a)** Menu board of the Pepblock Builder VR game. **(b)** A 3D visualisation of an in-game peptide, with *in silico* parameters on the right panel. **(c)** In-game challenge displaying the required final output in neon blue. **(d)** Structure manipulation by bending and snap-fitting into the required final output.

BOX 1 | Controls and gestures implemented for Pepblock Builder VR.

Button/action	Assigned function
Grip (L) (R)	Grab
Thumbstick (L)	Move around in VR
Thumbstick (R)	Turn around in VR
Trigger(R)	Click
Oculus (R)	Toggle Menu
Throw (action)	Merges structure in hand with the structure in display
Grip (L)/(R) + Thumbstick (L)/(R)	Moves the structure only* toward or away from the user

the VR application, the users can make combinations of these building blocks or even combine different protein structures from the PDB server. For basic experience, a certain number of permutations and combinations of the four elementary structures present in the front panel of the UI are pre-modelled and stored in the application library. Thus, when the user makes a combination of proteins using elementary shapes up to three levels, the Pepblock Builder VR instantly shows the resultant structure. However, just as the LEGO analogy, the complexities and permutations and combinations of possible structures that can be made are infinite. Thus, it would be impractical to have a resultant file for every possible protein structure made from the elementary structures. To tackle this, the Pepblock Builder VR was linked to a stand-alone protein modelling application. In the backend, the resultant structure is modelled and displayed in the Pepblock Builder VR UI when ready. In our case, I-Tasser standalone application was used for protein modelling. However, this is a highly time-consuming process and dramatically increases the wait times for complex structures. Once a user gets back the resultant structure, the user will be able to store the structure locally and be able to share the structure with any other user through in-game file transfer. This enables the creation of novel protein structures in a gamified and experiential manner and without the need to work with the corresponding amino acid sequences of the proteins. As a design tool, the Pepblock Builder VR shows a promising potential to bring protein design to a broader non-technical audience.

User Response and Usability

Twenty random users with a combination of low, medium and professional knowledge on protein were asked to evaluate the Pepblock Builder VR in aspects such as usability, ease of learning about proteins, satisfaction with the VR UI and the potential for the Pepblock Builder VR as a scientific tool. The user response is shown in **Figure 7**.

The response shows that the vast majority of the users found the Pepblock Builder VR as a very interactive and useful bio-edutainment tool. The distribution of experiencing different levels of difficulty to play the game could be due to the varied level

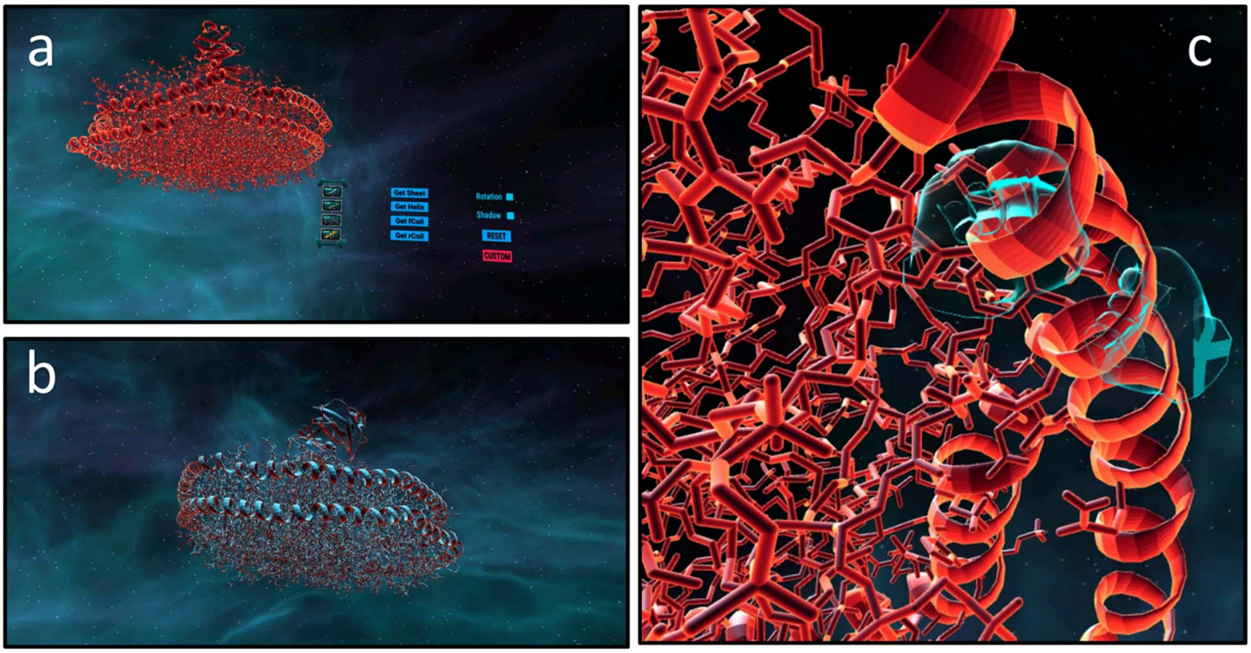


FIGURE 6 | Custom protein visualisation in the Pepblock Builder VR. Protein complex "6CLZ–PDB ID" was downloaded from the RCSB PDB website and was automatically converted to a.DAE file on the Pepblock Builder VR. **(a)** Showing the VR interface for importing custom protein. **(b)** Showing 6CLZ in the Pepblock Builder VR environment. **(c)** Outer helices of the 6CLZ nano disc being rotated by virtual hands.

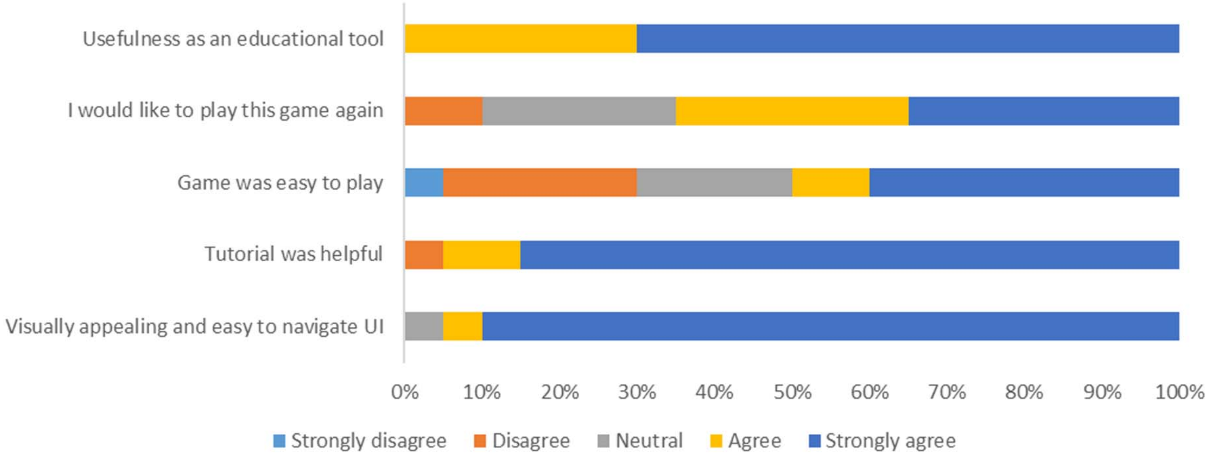


FIGURE 7 | User response toward the Pepblock Builder VR. Twenty users were asked to take an anonymous survey after experiencing the game and custom protein visualisation in virtual reality.

of prior knowledge in proteins. However, most users found the tutorial (including the cutscene) very helpful in understanding the context and the background of the game.

DISCUSSION AND OUTLOOK

We developed the Pepblock Builder VR by blending the concepts of protein design, 3D visualisation and VR and exploiting the merits of gamification. Thorough care was taken during every iteration of the tool to provide an interactive interface to users of all levels. Considering that not all users would have the availability of HMD equipment for the VR environment Pepblock Builder VR package, we also packed the entire gamified learning experience into a desktop version. The minimum requirements to run the Pepblock Builder VR are described in **Figure 8**.

The existing VR tools for molecular visualisation are often complicated to navigate, and some tools also require basic

Minimum system requirements

Pepblock Builder VR

- Oculus Rift hardware set-up
- Handheld controllers
- Unity game engine installed
- A computer with
  - Windows 7 or higher/ Mac OS 10 or higher
  - Oculus software installed
  - 3 USB 2.0 or higher
  - 4 GB or higher memory
  - Graphics 2GB or higher

Pepblock Builder VR Standalone version

- Optical mouse
- A computer with
  - Windows 7 or higher/ Mac OS 10 or higher
  - 3 x USB 2.0 or higher
  - 4 GB or higher memory
  - Graphics 2GB or higher

FIGURE 8 | Minimum requirements for running the Pepblock Builder VR.

programming skills. Compared to these, the Pepblock Builder VR has non-expert friendly, one-step installation procedure, simple navigation and workflow. The Pepblock Builder VR's interface is designed to interact and inform the user with a guided tutorial at every screen. The cutscene and the pop-up tutorials explain the game context and provide the users with in-game help wherever required.

The Pepblock Builder VR is enriched with multiple challenges in each level and edutains the users with fun facts about proteins at the end of each level. Such a gamified learning experience is unique to the Pepblock Builder VR. Gamification principles have not been extensively used priorly in any other molecular visualisation tools except in Fold.it. The gamification of the Pepblock Builder VR adopts a LEGO-style modular protein design. Proteins are versatile biomolecules. Fusion proteins (proteins made by combining two or more full/partial proteins) and engineered versions of natural proteins have been revolutionising various fields such as biomedicine, materials technology and food processing. The concepts of protein design could be compared to a LEGO-style modular approach with a twist. For example, a combination of structures A [Helix] and B [Coil] may not be [A + B], i.e., Helix attached to a coil, and in most cases may result in a completely new structure C. This non-linear nature of combining two protein structures is difficult to understand without some basic theoretical knowledge on free energy and stereochemistry. This gets more complicated when protein design is introduced. In such cases, deploying gaming-based learning provides a solution.

Gaming principles involve core elements such as concept building, world realisation and problem solving. Users learn and adapt to the in-game principles as they progress through levels. This slow introduction of world rules and self-adapted learning of concepts is an effective alternative way compared to the traditional classroom-based learning of protein design. The process of designing solutions, modelling ideas, building strategies and testing the outputs is a form of the "design, model, build and test" approach, a cornerstone of synthetic biology (Agapakis, 2021; Bueso and Tangney, 2021). Combined with a gamified protein design interface, an immersive VR experience and a LEGO-style protein building interface, the Pepblock Builder helps in the understanding of protein folding concepts for both technical and non-technical users. This was successfully observed from the user experience survey. Nearly 90% of the users found the Pepblock Builder VR easy to navigate and visually appealing. Over 70% of the users strongly agreed that the Pepblock Builder VR is a useful educational tool.

The Pepblock Builder VR could also be used as a simple visualisation tool. Any protein structure could be visualised in VR using the Pepblock Builder VR. Protein complexes and small molecules can also be imported into the VR interface. The Pepblock Builder VR can fetch *in silico* parameter data for the imported structure by interacting with online servers and in-built scripts.

The current version of the Pepblock Builder VR is a proof-of-concept tool with several limitations. One of the key limitations of the game is the time consumed while modelling the new combinations of the protein structure. The current game database has limited permutations and combinations of the four basic shapes provided in the left panel. Levels 1–10 rely on this in-game database. As the levels progress, the demand of newer combinations will rise. This requires protein modelling. Currently, with a lab-grade computer, modelling a 100–200 amino acid chain would take 5 h and 12–15 h for 500 AA long protein, depending on the structural complexity. Real-time modelling of such structures would be challenging to achieve without long wait times. Improving the time consumed for protein modelling is a bottleneck that we aim to improve in the next iterations of the game. In addition to this, to maintain the attention of the users during the wait periods, creative subtasks and facts-based edutainment levels will be implemented.

In the current version, game levels only up to level 10 are included. We plan to introduce packages with more game levels and new challenges in the near future. The first 10 levels of the Pepblock Builder VR have an in-game shadow to direct users toward the end goal of each level. The *in silico* parameters displayed in the right panel change in real time as the protein structure is being modified. This feature becomes the primary guide for level 11 onward. The users would be challenged to make/design a structure for a defined set of *in silico* parameters. This expands the potential of the Pepblock Builder VR from a bio-edutainment and visualisation tool to a citizen-based protein design interface. In future, cloud-based libraries for user-designed proteins would be established, enabling multiplayer capabilities and providing "share and build" features.

The Pepblock Builder VR opens a new avenue in protein design. With advancements in parallel computing, increasing computational power and improving graphics, the future of bringing protein design to the masses looks promising.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

The authors acknowledge relevant support from Breakthrough Cancer Research, Science Foundation Ireland (18/SPP/3522,15/CDA/3630 and 18/CRT/6222) and the Health Research Board (MRCG2016-25).

REFERENCES

Agapakis, C. M. (2021). *Designing Synthetic Biology | ACS Synthetic Biology*. Available online at: https://pubs.acs.org/doi/abs/10.1021/sb4001068?src=recsys (accessed February. 27, 2021).

Anderson-Lee, J., Fisker, E., Kosaraju, V., Wu, M., Kong, J., Lee, J., et al. (2016). Principles for Predicting RNA Secondary Structure Design Difficulty. *J. Mol. Biol.* 428, 748–757. doi: 10.1016/j.jmb.2015.11.013

Anikina, O. V., and Yakimenko, E. V. (2015). Edutainment as a Modern Technology of Education. *Procedia-Soc. Behav. Sci.* 166, 475–479. doi: 10.1016/j.sbspro.2014.12.558

Bank, R. P. D. (2021). *PDB Statistics: Overall Growth of Released Structures Per Year*. Available online at: https://www.rcsb.org/stats/growth/growth-released-structures (accessed February. 27, 2021).

Brainflight (2021). *Brainflight*. Available online at: http://brainflight.org/ (accessed March. 21, 2021).

Bueso, Y. F., and Tangney, M. (2021). *Synthetic Biology in the Driving Seat of the Bioeconomy: Trends in Biotechnology*. Available online at: https://www.cell.com/trends/biotechnology/fulltext/S0167-7799(17)30021-5 (accessed February. 27, 2021).

Cai, Y. Y., Indhumathi, C., Chen, W. Y., and Zheng, J. M. (2008). "VR Bio X Games," in *Transactions on Edutainment I*, eds Z. Pan, A. D. Cheok, W. Müller, and A. El Rhalibi (Berlin: Springer), 278–287.

Cai, Y., Lu, B., Fan, Z., Indhumathi, C., TeckLim, K., WernChan, C., et al. (2006). Bio-edutainment: Learning life science through X gaming. *Comput. Amp Graph.* 30, 3–9.

Cai, Y., Lu, B., Zheng, J., and Li, L. (2021). *Immersive protein gaming for bio edutainment, 2006*. Available online at: https://journals.sagepub.com/doi/10.1177/1046878106293677 (accessed February. 27, 2021).

Cooper, S., Sterling, A. L. R., Kleffner, R., Silversmith, W. M., and Siegel, J. B. (2018). "Repurposing citizen science games as software tools for professional scientists," in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, (New York, NY), 1–6. doi: 10.1145/3235765.3235770

DeFanti, T. A., Dawe, G., Sandin, D., Schulze, J., Otto, P., Girado, J., et al. (2009). The StarCAVE, a third-generation CAVE and virtual reality OptIPortal. *Future Gener. Comput. Syst.* 25, 169–178. doi: 10.1016/j.future.2008.07.015

Global Edutainment Market Growth (2021). *Global Edutainment Market Growth, Analysis up to 2028*. Available online at: https://www.futuremarketinsights.com/press-release/edutainment-market (accessed February. 27, 2021).

Goddard, T. D., Brilliant, A. A., Skillman, T. L., Vergenz, S., Tyrwhitt-Drake, J., Meng, E. C., et al. (2018a). Molecular Visualization on the Holodeck. *J. Mol. Biol.* 430, 3982–3996. doi: 10.1016/j.jmb.2018.06.040

Goddard, T. D., Huang, C. C., Meng, E. C., Pettersen, E. F., Couch, G. S., Morris, J. H., et al. (2018b). UCSF ChimeraX: Meeting modern challenges in visualization and analysis. *Protein Sci.* 27, 14–25. doi: 10.1002/pro.3235

Good, B. M., Loguercio, S., Griffith, O. L., Nanis, M., Wu, C., and Su, A. I. (2014). The Cure: Design and Evaluation of a Crowdsourcing Game for Gene Selection for Breast Cancer Survival Prediction. *JMIR Serious Games* 2:e3350. doi: 10.2196/games.3350

Huang, P.-S., Ban, Y., Florian, R., André, I., Vernon, R., Schief, W., et al. (2011). RosettaRemodel: a generalized framework for flexible backbone protein design. *PLoS One* 6:e24109. doi: 10.1371/journal.pone.0024109

Huang, P.-S., Boyken, S. E., and Baker, D. (2016). The coming of age of de novo protein design. *Nature* 537, 320–327. doi: 10.1038/nature19946

Humphrey, W., Dalke, A., and Schulten, K. (1996). VMD: visual molecular dynamics. *J. Mol. Graph.* 14, 33–38. doi: 10.1016/0263-7855(96)00018-5

Indhumathi, C., Cai, Y. Y., Cao, C. R., Lu, B. F., and Zheng, J. M. (2007). Virtual reality prototyping of bio-molecules. *Virtual Phys. Prototyp.* 2, 37–49. doi: 10.1080/17452750601170316

Jmol. (2021). *Jmol: an open-source Java viewer for chemical structures in 3D*. Available online at: http://jmol.sourceforge.net/ (accessed February. 27, 2021).

Kleffner, R., Flatten, J., Leaver-Fay, A., Baker, D., Siegel, J. B., Khatib, F., et al. (2017). Foldit Standalone: a video game-derived protein structure manipulation interface using Rosetta. *Bioinformatics* 33, 2765–2767. doi: 10.1093/bioinformatics/btx283

Koepnick, B., Flatten, J., Husain, T., Ford, A., Silva, D. A., Bick, M. J., et al. (2019). De novo protein design by citizen scientists. *Nature* 570:7761. doi: 10.1038/s41586-019-1274-4

Kwak, D., Kam, A., Becerra, D., Zhou, Q., Hops, A., Zarour, E., et al. (2013). Open-Phylo: a customizable crowd-computing platform for multiple sequence alignment. *Genome Biol.* 14:R116. doi: 10.1186/gb-2013-14-10-r116

Li, H., Leung, K.-S., Nakane, T., and Wong, M.-H. (2014). iview: an interactive WebGL visualizer for protein-ligand complex. *BMC Bioinform.* 15:56–56. doi: 10.1186/1471-2105-15-56

Norrby, M., Grebner, C., Eriksson, J., and Boström, J. (2015). Molecular Rift: Virtual Reality for Drug Designers. *J. Chem. Inf. Model.* 55:554.s doi: 10.1021/acs.jcim.5b00544

Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C., et al. (2004). UCSF Chimera–a visualization system for exploratory research and analysis. *J. Comput. Chem.* 25, 1605–1612. doi: 10.1002/jcc.20084

PyMOL (2021). *PyMOL | pymol.org*. Available online at: https://pymol.org/2/ (accessed February. 27, 2021).

Richardson, D. C., and Richardson, J. S. (2002). Teaching molecular 3-D literacy. *Biochem. Mol. Biol. Educ.* 30, 21–26. doi: 10.1002/bmb.2002.494030010005

Sayle, R. A., and Milner-White, E. J. (1995). RASMOL: biomolecular graphics for all. *Trends Biochem. Sci.* 20, 374–376. doi: 10.1016/S0968-0004(00)89080-5

Wood, C. W., Heal, J. W., Thomson, A. R., Bartlett, G. J., Ibarra, A. Á, Brady, R. L., et al. (2017). ISAMBARD: an open-source computational environment for biomolecular analysis, modelling and design. *Bioinforma. Oxf. Engl.* 33, 3043–3050. doi: 10.1093/bioinformatics/btx352

Yeh, C.-T., Brunette, T. J., Baker, D., McIntosh-Smith, S., and Parmeggiani, F. (2018). Elfin: An algorithm for the computational design of custom three-dimensional structures from modular repeat protein building blocks. *J. Struct. Biol.* 201, 100–107. doi: 10.1016/j.jsb.2017.09.001

Zhang, J. F., Paciorkowski, A. R., Craig, P. A., and Cui, F. (2019). BioVR: a platform for virtual reality assisted biological data integration and visualization. *BMC Bioinformatics* 20:78. doi: 10.1186/s12859-019-2666-z

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers
in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 15 June 2021
doi: 10.3389/fbioe.2021.687426

Protein Secondary Structure Prediction With a Reductive Deep Learning Method

Zhiliang Lyu[1†], Zhijin Wang[1†], Fangfang Luo[1], Jianwei Shuai[2,3] and Yandong Huang[1*]*

[1] College of Computer Engineering, Jimei University, Xiamen, China, [2] Department of Physics and Fujian Provincial Key Laboratory for Soft Functional Materials Research, Xiamen University, Xiamen, China, [3] National Institute for Data Science in Health and Medicine, and State Key Laboratory of Cellular Stress Biology, Innovation Center for Cell Signaling Network, Xiamen University, Xiamen, China

OPEN ACCESS

Edited by:
Zhibin Lv,
University of Electronic Science and Technology of China, China

Reviewed by:
Han Wang,
Northeast Normal University, China
Xiangzheng Fu,
Hunan University, China

****Correspondence:***
Yandong Huang
yandonghuang@jmu.edu.cn
Jianwei Shuai
jianweishuai@xmu.edu.cn

†These authors have contributed equally to this work and share first authorship

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *29 March 2021*
Accepted: *26 April 2021*
Published: *15 June 2021*

Citation:
Lyu Z, Wang Z, Luo F, Shuai J and Huang Y (2021) Protein Secondary Structure Prediction With a Reductive Deep Learning Method. Front. Bioeng. Biotechnol. 9:687426. doi: 10.3389/fbioe.2021.687426

Protein secondary structures have been identified as the links in the physical processes of primary sequences, typically random coils, folding into functional tertiary structures that enable proteins to involve a variety of biological events in life science. Therefore, an efficient protein secondary structure predictor is of importance especially when the structure of an amino acid sequence fragment is not solved by high-resolution experiments, such as X-ray crystallography, cryo-electron microscopy, and nuclear magnetic resonance spectroscopy, which are usually time consuming and expensive. In this paper, a reductive deep learning model MLPRNN has been proposed to predict either 3-state or 8-state protein secondary structures. The prediction accuracy by the MLPRNN on the publicly available benchmark CB513 data set is comparable with those by other state-of-the-art models. More importantly, taking into account the reductive architecture, MLPRNN could be a baseline for future developments.

Keywords: protein secondary structure, deep learning, multilayer perceptron, recurrent neural network, sequence profile

1. INTRODUCTION

Proteins are biomacromolecules that function in various life processes, many of which have been found as drug targets of human diseases (Huang et al., 2016; Li et al., 2021). The syntheses of proteins as long polypeptide chains or primary sequences take place in the ribosomes. Released from the ribosomes, the chains fold spontaneously to produce functional three-dimensional structures or tertiary structures (Anfinsen et al., 1961), which are usually determined by experiments, including X-ray crystallography, cryo-electron microscopy, and nuclear magnetic resonance spectroscopy. However, these experiments are often time consuming and expensive, which to a large extent explains the gap between the number of protein structures (∼150,000) deposited in the Protein Data Bank (PDB) (Berman et al., 2002) and that of sequences (∼140,000,000) stored in the UniProtKB/TrEMBL database (The UniProt Consortium, 2017, 2018). Therefore, it is of importance to develop efficient computational methods for protein structure prediction. The three-dimensional structure of a protein is determined most by its amino acid sequence (Baker and Sali, 2001), indicating the possibility of theoretical prediction of a protein structure from its amino acid sequence.

Protein secondary structures are characterized as local structures that are stabilized by hydrogen bonds on the backbone and considered as the linkages between primary sequences and tertiary structures (Myers and Oas, 2001; Zhang, 2008; Källberg et al., 2012). According to the distinct hydrogen bonding modes, generally three types of secondary structures have been identified, namely helix (H), strand (E), and coil (C), where the helix and strand structures are most common in nature (Pauling et al., 1951). Later in 1983, a finer characterization of secondary structures was proposed. In the new classification calculated by DSSP algorithm, previous 3 states are extended to 8 states, including α-helix (H), 3_{10} helix (G), π-helix (I), β-strand (E), β-bridge (B), β-turn (T), bend (S), and loop or others (C) (Kabsch and Sander, 1983), among which the α-helix and β-strand are the principal structure features.

The 3-state or Q3 prediction problem has been extensively studied since 1974 (Chou and Fasman, 1974). As summarized by Stapor and coworkers, the computational models reported after 2007 can provide the prediction accuracy of 80% and above (Smolarczyk et al., 2020). Until 2018, the theoretical limit 88% of the Q3 protein secondary structure prediction was achieved first by Lu group (Zhang et al., 2018). At the same time, it is noticed that the 8-state or Q8 prediction would provide more valuable information. For instance, π-helix is found abundant and associated with activities in some special proteins (Cooley et al., 2010). As a result, over the few years many efforts have been made, trying to solve the Q8 prediction problem, which is much more complicated and challenging (Li and Yu, 2016; Wang et al., 2016; Fang et al., 2017; Heffernan et al., 2017; Zhang et al., 2018; Krieger and Kececioglu, 2020; Uddin et al., 2020; Guo et al., 2021) If not otherwise specified, the models discussed in this paper are non-template based. The Q8 prediction accuracy has reached 70% and at present the best record is 77.73% (Uddin et al., 2020). Thus, there is still a deviation of about 10% from the theoretical limit of 88% (Rost et al., 1994).

Over the past few decades, a variety of state-of-the-art methods have been developed to improve Q3 or Q8 prediction accuracy and most progresses are contributed by machine learning based models (Li and Yu, 2016; Wang et al., 2016; Fang et al., 2017; Heffernan et al., 2017; Zhang et al., 2018; Krieger and Kececioglu, 2020; Uddin et al., 2020; Guo et al., 2021) So far as we know, the predictive power of a machine learning model is governed mainly by two elements, namely feature representation and algorithm. For instance, the introduction of sequence evolutionary profiles from multiple-sequence alignment (Rost and Sander, 1993), such as position-specific scoring matrices (PSSM) (Jones, 1999), improves prediction accuracy significantly (Zhou and Troyanskaya, 2014). In addition to PSSM, either the hidden Markov model (HMM) profile (Guo et al., 2021) or amino acid parameters (Zhang et al., 2018) can also contribute to the improvement of prediction accuracy. As to a machine learning algorithm, the major task is to capture either local or non-local dependencies from the input features using different neural network architectures. For instance, a specific neural network, namely convolutional neural network (CNN) (LeCun et al., 1998), is successful in capturing short-range features. At the same time, the recurrent neural network (RNN) equipped with bidirectional gate current unit (BGRU) (Cho et al., 2014) or long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) can be used to capture long-range dependencies. CNN and RNN architectures were integrated for the first time in the DCRNN model to predict protein secondary structures (Li and Yu, 2016; Zhang et al., 2018). Some models employ different deep learning architectures, such as the deep conditioned neural field (DeepCNF) (Wang et al., 2016) and the deep inception-inside-inception network (Deep3I) (Fang et al., 2017; Uddin et al., 2020). In particular, the model SAINT that incorporates self-attention mechanism and Deep3I provides up-to-date the best Q8 prediction accuracy (Uddin et al., 2020).

Noting that as the neural network architecture gets more complex or deeper, the number of parameters grows. In this work, a reductive neural network architecture MLPRNN has been proposed that include a two-layer stacked bidirectional gated recurrent unit (BGRU) block capped by two multilayer perceptrons (MLP) at both sides, like a sandwich. Encouragingly, the prediction accuracy for Q3 and Q8 reach 83.32 and 70.59%, respectively, comparable with other state-of-the-art methods developed recently. More importantly, taking into account the reductive architecture, MLPRNN would provide an extensible framework for future developments.

2. METHODS AND MATERIALS

2.1. Data Sets

In this work, two publicly available data sets, CB6133-filtered and CB513 (Zhou and Troyanskaya, 2014), which have been widely applied in protein secondary structure prediction (Li and Yu, 2016; Fang et al., 2017; Zhang et al., 2018; Guo et al., 2021), were used to train and test the new model, respectively. The CB6133-filtered is the result of removing the sequences that have >25% identity with the CB513 and the redundancy with the CB513 from the original CB6133. As expected, the distributions of 8 states with respect to the CB6133-filtered and CB513 are similar (**Supplementary Figure 6**).

2.1.1. CB6133-Filtered

An open-source protein sequence data set, namely CB6133-filtered, was employed for training in this work (Zhou and Troyanskaya, 2014). CB6133-filtered is a large non-homologous sequence and structure data set that contains 5,600 training sequences. This data set was produced with the PISCES Cull PDB server, a public server for culling sets of protein sequences from the Protein Data Bank (PDB) by the sequence identity and structural quality criteria (Wang and Dunbrack, 2003). Notably, the data set was created with better than 2.5Å resolution while sharing less than 30% identity.

2.1.2. CB513

The testing data set CB513 was introduced by Cuff and Barton (Cuff and Barton, 1999, 2000). Noting that the length of one sequence is longer than the maximal of 700, this sequence has been split into two overlapping sequences. As a result, CB513 contains 514 sequences. Both CB6133-filtered and CB513 data sets can be downloaded via Zhou's website.

2.2. Input Features

2.2.1. PSSM Profile

Statistically, homologous proteins often have similar secondary structures. Thus, all homologous proteins can be grouped into a family through the multiple sequence alignment (MSA) with a fitting cutoff (Sander and Schneider, 1991). Then the approximate structure of the family can be predicted. Apparently, the MSA gives much more structural information than one single sequence (Rost and Sander, 1993). One of the most popular position-specific profile of proteins is the PSSM (Jones, 1999), which can be produced by the PSI-BLAST algorithm (Altschul et al., 1997). The PSSM dimension of a sequence is $N \times S$, where N and S denote the types of amino acids and the length of the sequence, respectively. Normally, N is 20 that corresponds to the 20 standard amino acid types. Here, one additional type, marked as X, was added to the PSSM profile to represent non-standard amino acids. Thus, N is 21 instead of 20 for the PSSM profile. According to the PSI-BLAST, each position of amino acids gets a score of hit that denotes the appropriate probability of the amino acid staying in this position solidly. For instance, if the score of the hit is high, a position is supposed to be conserved. Otherwise, the position is not likely a conserved site (Gribskov et al., 1987; Jeong et al., 2010). Usually, a sigmoid function is applied to restrain the scores of the hits that range from 0 to 1 (Jones, 1999).

2.2.2. HMM Profile

Recently, it has been demonstrated that the combination of HMM and PSSM profiles as input of the model DNSS2 can improve the Q8 prediction accuracy by about 2% (Guo et al., 2021). Thus, in this work, we follow the scheme above and the PSSM and HMM profiles were used as input. The HMM profile was calculated with the HHblits (Remmert et al., 2012), a software that can convert amino acid sequences into hidden Markov model profiles by searching specific databases iteratively. The database used in this work is the publicly available *uniclust30_2016_03.tgz*. The columns in the HMM profile correspond to the 20 amino acid types. In each column, a substitution probability is provided based on its position along the protein sequence (Smolarczyk et al., 2020). Finally, the values generated by the HHblits were transformed to the linear probabilities, which can be formulated as follows:

$$p = 2^{-N/1000} \tag{1}$$

where N denotes the score number from the profile (Sharma et al., 2016). Compared to the sequence-search tool PSI -BLAST, HHblits is faster because of its discretized-profile prefilter. Also, HHBlits is more sensitive than PSI-BLAST (Remmert et al., 2012).

2.3. Model Design

The reductive model MLPRNN proposed in this study is composed by one BGRU and two MLP blocks. In this section, MLP and BGRU will be introduced separately. Followed is the explanation in details of the overall architecture.

2.3.1. MLP

The multi-layer perceptron (MLP) is a reductive neural network with at least three layers, namely an input layer, a hidden layer, and an output layer. Taking the three-layer MLP exploited in this study as an example, as illustrated in **Figure 1**, each neuron at the hidden layer integrates the messages from all input nodes and spreads the integrated message to all neurons at the output layer. A linear function is used to adjust the number of neurons at each layer. Each neuron need to work with a non-linear activation function, such as Rectified Linear Unit (ReLU), and a dropout method.

2.3.2. BGRU

In this study, the bidirectional gate current units (BGRUs) were used to capture long-range dependencies in the amino acid sequences. Assuming the number of hidden units is k and the input of a GRU(t) is (l_t, h_{t-1}). The activated reset gate r_t, update gate u_t, internal memory cell $\widetilde{h}_t$, and GRU output $h_t(\in \mathbb{R}^k)$ can be expressed as follows:

$$r_t = \sigma(W_{lr} l_t + W_{hr} h_{t-1} + b_r) \tag{2}$$

$$u_t = \sigma(W_{lu} l_t + W_{hu} h_{t-1} + b_u) \tag{3}$$

$$\widetilde{h}_t = tanh(W_{l\widetilde{h}} l_t + W_{h\widetilde{h}}(r_t \odot h_{t-1} + b_{\widetilde{h}})) \tag{4}$$

$$h_t = u_t \odot h_{t-1} + (1 - u_t) \odot \widetilde{h}_t \tag{5}$$

where W_{lr}, W_{hr}, W_{lu}, W_{hu}, $W_{l\widetilde{h}}$, and $W_{h\widetilde{h}}$ $(\in \mathbb{R}^{3q \times k})$ denote weight matrices. b_r, b_u, and $b_{\widetilde{h}}$ $(\in \mathbb{R}^k)$ are bias terms. $\odot$, σ, and *tanh* stand for element-wise multiplication, sigmoid, and hyperbolic functions, respectively (Li and Yu, 2016). As illustrated in the inset of **Figure 1**, each GRU contains one input and one output. A BGRU layer, such as BGRU 1 in **Figure 1**, not only learns input features from head to tail, but also tail to head, so as to catch the dependencies at both sides. Thus, a BGRU need read input features twice. In the end, outputs of two GRU chains are merged together as the final output.

2.3.3. Overview of MLPRNN

Figure 1 illustrates the data stream of an amino acid in the sequences and the other dimension perpendicular to the plot is the amino acid sequences. As illustrated in **Figure 1**, MLPRNN has a sandwich like architecture where a two-layer stacked BGRU block is capped by two MLP blocks at both sides. Both MLP blocks have one hidden layer. In specific, 41-dimensional features are taken as the input of the first MLP block. The dimensions of the input, hidden, and output layers in the first MLP block are 41, 256, and 512, respectively. The BGRU block is fed with the 512-dimensional output of the first MLP. The BGRU block is followed by the other MLP block with one hidden layer too. The dimensions of the input, hidden, and output layers are 512, 256, and 9, respectively. Finally, the prediction is made by a softmax unit fed by the output of the second MLP block. The dimensions of the hidden and output layers in the MLP blocks are selected based on the prediction accuracy. As shown

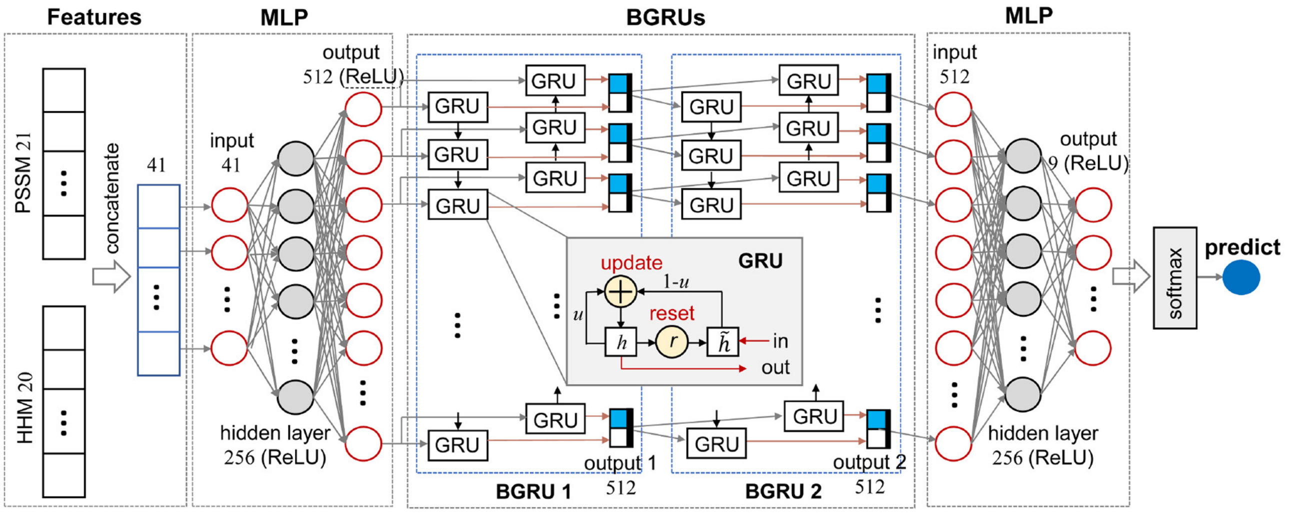


FIGURE 1 | Schematic diagram of the MLPRNN model.

in **Supplementary Table 1**, the combination of the dimensions 256 and 512 give not only the best Q8 prediction, but also the fastest convergence. From **Supplementary Table 2**, one can see that the model with two-layer stacked BGRU block gives best performance in view of accuracy as well as efficiency. For instance, the models with respect to two-layer and three-layer stacked BGRU blocks give similar accuracies, but the former has less parameters. Thus, the two-layer stacked BGRU block is chosen in this study.

2.4. Implementation Details

In all experiments, the optimizer named Adam was used during the training to calculate and update the parameters of the model. The default original learning rate is set 0.001, which decreases every 10 epochs with the rate of 0.997. All sequences were padded with zero if the sequence length is shorter than 700. As a consequence, zero could be learned by the model, which is undesired. To remove the effect of the zero class, the Multiple Cross-Entropy Loss function was employed, which is based on the cross-entropy loss function. The weight constraint of dropout with the parameter $p = 5$ was applied to avoiding over fitting by BGRUs and the tails of MLPs. Our experiments were implemented under the PyTorch (version 1.7.1) environment and the model was trained on a single NVIDIA Titan RTX GPU with 24 Gigabyte (GB) memory. Each experiment in this work was trained and tested for at least 3 times and the best result was taken as the final solution. In this work, the average of the loss over the last 10 epochs was used to determine at which epoch the convergence was reached for the testing set.

2.5. Performance Evaluation

The Q Score formulated as Equation (6) has been widely used to examine protein secondary structure predictions. In brief, it measures the percentage of residues for which the predicted secondary structures are correct (Wang et al., 2016).

$$Q_m = 100\% \times \frac{\sum_{i=1}^{m} N_{corr}(i)}{N} \tag{6}$$

where m indicates the number of classes. $m = 3$ and $m = 8$ correspond to Q3 and Q8 predictions, respectively (Lee, 2006). $N_{corr}(i)$ is the number of correctly predicted residues for state i and N is the total number of residues.

3. RESULTS AND DISCUSSION

3.1. Prediction Accuracy

Q3 and Q8 prediction accuracy have been estimated by the proposed model MLPRNN and compared with the values by another 5 state-of-the-art methods that also used CB513 for testing. Here Q8 is transformed to Q3 by treating 3_{10}-helix and π-helix as α-helix (H) and merging β-bridge (B) to β-strand (E). As to the rest, turn (T) and bend (S) are treated as coil (C). As illustrated in **Table 1**, the prediction accuracy for either Q3 or Q8 by MLPRNN is at the same level with other state-of-the-art methods. In particular, the Q8 prediction accuracy obtained by the new model is about 1 and 3% lower than those given by CRRNN (Zhang et al., 2018) and DNSS2 (Guo et al., 2021), respectively. Here, the DNSS2 integrates 6 deep learning architectures, which is much more complex than the present MLPRNN. In addition to the PSSM and HMM profiles, another three input features were utilized in the DNSS2 model (Guo et al., 2021). With respect to CRRNN, the training set TR12148 applied by this model is about twice larger than the CB6133-filtered used in this work (Zhang et al., 2018). Thus, the present MLPRNN could be improved with more input features such as the ones introduced by DNSS2 or a larger training dataset like the TR12148. It should be noted that MLPRNN and DNSS2 share the same method of mapping Q8 to Q3. Although CRRNN and DeepCNF use another method for the transformation. In specific,

TABLE 1 | Q3 and Q8 prediction accuracy (%) comparison.

Method	References	Q3	Q8
DeepCNF	Wang et al., 2016	82.30	68.30
MUFOLD-SS	Fang et al., 2017	82.98	71.05
BGRUCB	Drori et al., 2018	82.85	70.10
CRRNN	Zhang et al., 2018	85.30	71.40
DNSS2	Guo et al., 2021	82.56	73.36
MLPRNN		83.32	70.59

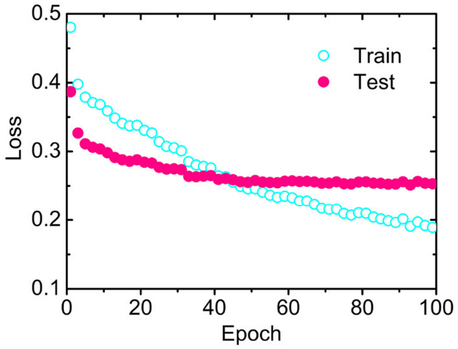


FIGURE 2 | Losses as a function of epoch by MLPRNN for the training (open circles) and testing (solid circles) data sets, respectively.

α-helix (H), β-strand (E), and the rest 6 states in Q8 form the 3 classes of Q3, respectively. It has been reported that the selection of the transformation method from Q8 to Q3 can influence prediction performance to some extent (Cuff and Barton, 1999). Indeed, replacing the present method of converting Q8 to Q3 with the one employed by CRRNN, the prediction accuracy of Q3 by MLPRNN increases from 83.32 to 85.38%, slightly higher than 85.30% by CRRNN.

3.2. Convergence Rate

The losses as a function of epoch for the training (CB6133-filtered) and testing (CB513) data sets, respectively, have been calculated to examine the convergence. As illustrated in **Figure 2**, the loss for CB513 drops from 0.39 to 0.30 within 6 epochs and stabilized or converged around 0.26 for another 38 epochs. The following two experiments have been designed, trying to explain the fast convergence of loss for CB513 by MLPRNN. First, the MLP blocks were removed from MLPRNN. As a result, the number of epochs required for loss convergence increases to 70 (**Supplementary Figure 1**), which is expected as BGRU is known as slow in learning when compared with other neural network architectures (Bradbury et al., 2016). Next, MLP was replaced with CNN, and the resulting convergence rate is similar with that by the original MLPRNN (see **Supplementary Figures 2, 3**). Thus, the sandwich-like reductive architecture itself is responsible for the fast loss convergence. It should be noted that MLP is more suitable than CNN for this model in terms of prediction accuracy, which will be discussed later.

TABLE 2 | Q8 prediction accuracy (%) with different input features.

Model	Q3	Q8
PSSM	82.27	69.50
HMM	80.51	62.49
PSSM+HMM	83.32	70.59

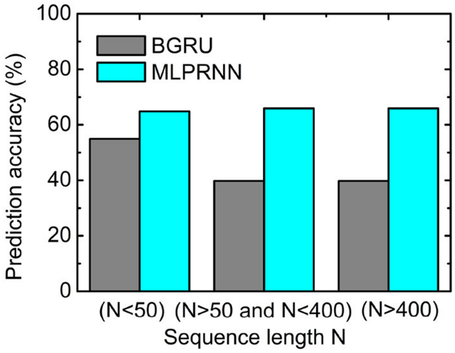


FIGURE 3 | Prediction accuracy obtained by the multilayer perceptron (MLP)-removed MLPRNN model (gray) and the original MLPRNN model (cyan) for three sequence length regions.

TABLE 3 | Q3 and Q8 prediction accuracy (%) where multilayer perceptrons (MLPs) in the MLPRNN are replaced by convolutional neural networks (CNNs).

Model	Q3	Q8
CNN (k = 1) BGRU	83.32	70.59
CNN (k = 3) BGRU	82.89	68.30
CNN (k = 7) BGRU	82.14	67.46

3.3. Feature Analysis

Feature representation is essential for the prediction of protein secondary structures. In this work, the input features are represented by the concatenation of PSSM and HMM profiles, both of which transfer the evolutionary information for amino acids in the sequences. Thus, it is of interest to examine the impacts of the two profiles separately. The loss convergence plots of the two experiments can be found in **Supplementary Figures 4, 5**. From **Table 2**, one can see that the prediction accuracy with PSSM profile is higher than that with HMM profile. In particular, the discrepancy is about 7% for Q8 prediction. However, when PSSM is combined with HMM, the prediction accuracy is improved by about 1% for both Q3 and Q8 predictions, implying that HMM profile is complementary to PSSM profile, which is consistent with the result obtained by the DNSS2 model (Guo et al., 2021).

Noting that the PSSM profile was generated by the PSI-BLAST, a profile-sequence alignment method, and the HMM profile was generated by the method HHblits that uses both profile-sequence alignment and profile–profile alignment. It has been suggested that the HHblits method is more sensitive to identify distant homologous sequences than the PSI-BLAST,

TABLE 4 | Prediction accuracy (%) for Q8 states.

Label	Types	Count	BGRU[a]	MLPRNN	MLPRNN (PSSM)[b]	MLPRNN (HMM)[c]	CNN(k = 3) BGRU[d]	CNN(k = 7) BGRU[e]
H	α-helix	405560	91.28	92.42	92.32	90.72	93.15	92.88
E	β-strand	255887	81.52	83.34	81.67	82.04	84.20	82.28
L	Coil	225493	64.48	68.34	64.97	67.22	71.22	71.36
T	Turn	132980	17.88	54.02	50.78	46.55	55.92	52.73
S	Bend	97298	6.73	26.83	27.91	0	0	0
G	3_{10}-helix	46019	1.50	25.73	29.92	0	0	0
B	β-bridge	12096	0	0	0	0	0	0
I	π-helix	209	0	0	0	0	0	0

[a] *MLPs are removed.*

[b] *Input features are represented by PSSM profile.*

[c] *Input features are represented by HMM profile.*

[d] *MLPs are replaced by CNNs with the kernel size k = 3.*

[e] *MLPs are replaced by CNNs with the kernel size k = 7.*

indicating different sensitivity and specificity between the two methods (Guo et al., 2021), which might explain the distinct performances between PSSM and HMM profiles found in the current protein secondary structure prediction. In specific, the PSI-BLAST method is perhaps more sensitive to the sequence homology of the datasets utilized in this work. In addition, the present HMM profile was generated based on a smaller sequence database, which might influence the accuracy of the HMM profile and the resulting prediction accuracy.

3.4. Model Analysis

The current reductive model MLPRNN is constructed by only a two-layer stacked BGRU block capped by two MLP blocks, facilitating detailed model analysis. To examine the impact of adding MLP blocks to both sides of BGRU block, the input data were trained with BGRU block alone and the resulting prediction accuracies are 73.22 and 61.95% for Q3 and Q8, respectively, about 10% lower than those by the original MLPRNN where MLP blocks are present. Apparently, the MLP blocks in the MLPRNN model are essential to the prediction.

Further, to investigate where the MLP-related improvement occurs, the sequences for testing were split into three groups according to the length N of a sequence. As illustrated in **Figure 3**, the prediction accuracy where N is larger than 50 is below 40%, about 15% lower than that where N is smaller than 50. When the MLP blocks are added, the prediction accuracies are all above 60% for the three length regions, indicating that MLP blocks could help capture very long-range dependencies. The experiment above highlights that the two MLP blocks are indispensable complementary to the BGRU block for protein secondary structure prediction.

CNNs have been used to couple with BGRUs for protein secondary structure prediction since 2016 (Li and Yu, 2016; Zhang et al., 2018). Therefore, it is of interest to see if the current framework works with CNNs too. In this experiment, MLPs in the MLPRNN model were replaced by CNNs where the kernel size k equals 3 or 7. Noting that a CNN with the kernel size $k = 1$ is equivalent to a MLP, MLPRNN is renamed as CNN(k = 1)BGRU in **Table 3**. From **Table 3**, one can see that the prediction accuracy reduces as the kernel size increases, which is more evident for Q8 prediction, demonstrating that MLPs match better with BGRUs than CNNs under the proposed reductive architecture.

Standard RNNs include LSTMs and GRUs. Thus, it is worth investigating the effect of replacing BGRUs with bidirectional LSTMs (BLSTMs). As presented in **Supplementary Table 2**, the BLSTMs show no impact on the prediction accuracy except for the reduced convergence rate, which is mainly due to the increased amount of parameters.

3.5. Prediction Accuracy for Individual Q8 States

Apart from the overall accuracy, the predictive precision for each class of Q8 would provide more useful information. Thus, the prediction accuracies for all Q8 states were calculated and listed in **Table 4** that includes the results by the MLPRNN model and the experiments mentioned above. Here, the labels are ordered based on the counts of 8 states in the training data set. It is evident that the prediction of T by BGRU is poor when compared with those by others, indicating that MLP or CNN blocks in the current framework are essential to predict the turn structure. Interestingly, only the MLPRNN model fed with at least PSSM profile is able to distinguish S or G from other states, though the prediction accuracy is still low.

From the third column of **Table 4**, one can see that the count of S or G type is much smaller than those with respect to the four most populated types, namely H, E, L, and T. Under such a limited number of samples, accurate feature extraction is essential for the prediction of S or G type. When CNNs are used, local features are extracted preliminarily at the convolution step before entering the neural network. Here, the range of the local features is determined by the kernel size. When the kernel size of 3 or above is used, some very local information, which are critical for the prediction of S or G type, could be missed during the convolution step. As a consequence, the following training in the neural network would be affected. In that case, the kernel size of 1, which is equivalent to MLP employed by the proposed MLPRNN, might be necessary.

From the prediction accuracies for individual Q8 states, it is found that HMM profile compensates PSSM profile by improving the prediction accuracies of H, E, L, and T types. Adding HMM profile to PSSM profile as input, however, reduces the prediction accuracies of the two less populated states, namely S and G. In association with the discussion on input features above, the poor prediction of either G or S type with the HMM profile alone as input might be due to the underlying effect of sequence homology.

The results above have provided two messages, which might be useful for future development. First, PSSM profile is better than HMM profile in representing bend and 3_{10}-helix states. Second, MLP is more suitable than CNN in predicting the two states.

4. CONCLUSION

In this study, we proposed a reductive deep-learning architecture MLPRNN for protein secondary structure prediction. Based on the benchmark CB513 data set, the prediction accuracy for either Q3 or Q8 by MLPRNN is comparable with those by other state-of-the-art methods, verifying the validity of this reductive model. From the comparative experiments, it is found that MLPs are non-trivial to the proposed model. First, MLPs contribute a lot to secondary structure prediction made by MPLRNN, especially at the long sequence length side. Besides, the reductive model performs better in the presence of MLPs instead of CNNs. The impact of input features have been studied too. It is revealed that, in contrast to PSSM profile, HMM profile fails in representing two less populated states, bend and 3_{10}-helix. In addition, the prediction of the two states fails too if the MLPs in the MLPRNN model are replaced with CNNs. Encouragingly, the original MLPRNN model in the presence of MLPs could capture features of the two states represented by PSSM profile. Finally, the MLPRNN model proposed in this study has provided a reductive and extensible deep learning framework, facilitating the incorporation of more sophisticated algorithms or new features in future for further improvement.

DATA AVAILABILITY STATEMENT

The code of MLPRNN and the relevant data can be downloaded from https://gitlab.com/yandonghuang/mlpbgru.

AUTHOR CONTRIBUTIONS

YH and ZW conceived the idea of this research. ZL and ZW performed the model implementation. ZL performed the data collection, training, and testing. ZL and FL performed the data analysis. YH, ZL, and JS wrote the manuscript. YH and JS supervised the research and reviewed the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by grants from the National Natural Science Foundation of China (Nos. 11804114, 11874310, and 12090052) and the National Key R&D Program of China (No. 2018YFD0901004).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fbioe.2021.687426/full#supplementary-material

REFERENCES

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., et al. (1997). Gapped blast and PSI-blast: a new generation of protein database search programs. *Nucleic Acids Res*. 25, 3389–3402. doi: 10.1093/nar/25.17.3389

Anfinsen, C. B., Haber, E., Sela, M., and White, F. Jr. (1961). The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proc. Natl. Acad. Sci. U.S.A*. 47, 1309–1314. doi: 10.1073/pnas.47.9.1309

Baker, D., and Sali, A. (2001). Protein structure prediction and structural genomics. *Science* 294, 93–96. doi: 10.1126/science.1065659

Berman, H. M., Battistuz, T., Bhat, T. N., Bluhm, W. F., Bourne, P. E., Burkhardt, K., et al. (2002). The protein data bank. *Acta Crystallogr. Sec. D Biol. Crystallogr*. 58, 899–907. doi: 10.1107/S0907444902003451

Bradbury, J., Merity, S., Xiong, C., and Socher, R. (2016). Quasi-recurrent neural networks. *arXiv [Preprint]*. arXiv:1611.01576.

Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*. doi: 10.3115/v1/W14-4012

Chou, P. Y., and Fasman, G. D. (1974). Prediction of protein conformation. *Biochemistry* 13, 222–245. doi: 10.1021/bi00699a002

Cooley, R. B., Arp, D. J., and Karplus, P. A. (2010). Evolutionary origin of a secondary structure: π-helices as cryptic but widespread insertional variations of α-helices that enhance protein functionality. *J. Mol. Biol*. 404, 232–246. doi: 10.1016/j.jmb.2010.09.034

Cuff, J. A., and Barton, G. J. (1999). Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins Struct. Funct. Bioinform*. 34, 508–519. doi: 10.1002/(SICI)1097-0134(19990301)34:4<508::AID-PROT10>3.0.CO;2-4

Cuff, J. A., and Barton, G. J. (2000). Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins Struct. Funct. Bioinform*. 40, 502–511. doi: 10.1002/1097-0134(20000815)40:3<502::AID-PROT170>3.0.CO;2-Q

Drori, I., Dwivedi, I., Shrestha, P., Wan, J., Wang, Y., He, Y., et al. (2018). High quality prediction of protein Q8 secondary structure by diverse neural network architectures. *arXiv [Preprint]*. arXiv:1811.07143.

Fang, C., Shang, Y., and Xu, D. (2017). Mufold-SS: protein secondary structure prediction using deep inception-inside-inception networks. *arXiv [Preprint]*. arXiv:1709.06165.

Gribskov, M., McLachlan, A. D., and Eisenberg, D. (1987). Profile analysis: detection of distantly related proteins. *Proc. Natl. Acad. Sci. U.S.A*. 84, 4355–4358. doi: 10.1073/pnas.84.13.4355

Guo, Z., Hou, J., and Cheng, J. (2021). Dnss2: improved *ab initio* protein secondary structure prediction using advanced deep learning architectures. *Proteins Struct. Funct. Bioinform*. 89, 207–217. doi: 10.1002/prot.26007

Heffernan, R., Yang, Y., Paliwal, K., and Zhou, Y. (2017). Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. *Bioinformatics* 33, 2842–2849. doi: 10.1093/bioinformatics/btx218

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput*. 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Huang, Y., Chen, W., Dotson, D., Beckstein, O., and Shen, J. (2016). Mechanism of ph-dependent activation of the sodium-proton antiporter nhaa. *Nat. Commun*. 7:12940. doi: 10.1038/ncomms12940

Jeong, J. C., Lin, X., and Chen, X.-W. (2010). On position-specific scoring matrix for protein function prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform*. 8, 308–315. doi: 10.1109/TCBB.2010.93

Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol*. 292, 195–202. doi: 10.1006/jmbi.1999.3091

Kabsch, W., and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers Origin. Res. Biomol*. 22, 2577–2637. doi: 10.1002/bip.360221211

Källberg, M., Wang, H., Wang, S., Peng, J., Wang, Z., Lu, H., et al. (2012). Template-based protein structure modeling using the raptorx web server. *Nat. Protoc*. 7, 1511–1522. doi: 10.1038/nprot.2012.085

Krieger, S., and Kececioglu, J. (2020). Boosting the accuracy of protein secondary structure prediction through nearest neighbor search and method hybridization. *Bioinformatics* 36(Suppl 1):i317–i325. doi: 10.1093/bioinformatics/btaa336

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Lee, J. (2006). Measures for the assessment of fuzzy predictions of protein secondary structure. *Proteins Struct. Funct. Bioinform*. 65, 453–462. doi: 10.1002/prot.21164

Li, X., Zhong, C., Wu, R., Xu, X., Yang, Z., Cai, S., et al. (2021). RIP1-dependent linear and nonlinear recruitments of caspase-8 and RIP3 respectively to necrosome specify distinct cell death outcomes. *Protein Cell* 1–19. doi: 10.1007/s13238-020-00810-x

Li, Z., and Yu, Y. (2016). Protein secondary structure prediction using cascaded convolutional and recurrent neural networks. *arXiv [Preprint]*. arXiv:1604.07176.

Myers, J. K., and Oas, T. G. (2001). Preorganized secondary Structure as an important determinant of fast protein folding. *Nat. Struct. Biol*. 8, 552–558. doi: 10.1038/88626

Pauling, L., Corey, R. B., and Branson, H. R. (1951). The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Natl. Acad. Sci. U.S.A*. 37, 205–211. doi: 10.1073/pnas.37.4.205

Remmert, M., Biegert, A., Hauser, A., and Söding, J. (2012). Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nat. Methods* 9, 173–175. doi: 10.1038/nmeth.1818

Rost, B., and Sander, C. (1993). Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proc. Natl. Acad. Sci. U.S.A*. 90, 7558–7562. doi: 10.1073/pnas.90.16.7558

Rost, B., Sander, C., and Schneider, R. (1994). Redefining the goals of protein secondary structure prediction. *J. Mol. Biol*. 235, 13–26. doi: 10.1016/S0022-2836(05)80007-5

Sander, C., and Schneider, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins Struct. Funct. Bioinform*. 9, 56–68. doi: 10.1002/prot.340090107

Sharma, R., Kumar, S., Tsunoda, T., Patil, A., and Sharma, A. (2016). Predicting MoRFs in protein sequences using HMM profiles. *BMC Bioinformatics* 17:504. doi: 10.1186/s12859-016-1375-0

Smolarczyk, T., Roterman-Konieczna, I., and Stapor, K. (2020). Protein secondary structure prediction: a review of progress and directions. *Curr. Bioinform*. 15, 90–107. doi: 10.2174/1574893614666191017104639

The UniProt Consortium (2017). Uniprot: the universal protein knowledgebase. *Nucleic Acids Res*. 45, D158–D169. doi: 10.1093/nar/gkw1099

The UniProt Consortium (2018). Uniprot: the universal protein knowledgebase. *Nucleic acids Res*. 46:2699. doi: 10.1093/nar/gky092

Uddin, M. R., Mahbub, S., Rahman, M. S., and Bayzid, M. S. (2020). Saint: self-attention augmented inception-inside-inception network improves protein secondary structure prediction. *Bioinformatics* 36, 4599–4608. doi: 10.1093/bioinformatics/btaa531

Wang, G., and Dunbrack, R. L. Jr. (2003). Pisces: a protein sequence culling server. *Bioinformatics* 19, 1589–1591. doi: 10.1093/bioinformatics/btg224

Wang, S., Peng, J., Ma, J., and Xu, J. (2016). Protein secondary structure prediction using deep convolutional neural fields. *Sci. Rep*. 6, 1–11. doi: 10.1038/srep18962

Zhang, B., Li, J., and Lü, Q. (2018). Prediction of 8-state protein secondary structures by a novel deep learning architecture. *BMC Bioinformatics* 19:293. doi: 10.1186/s12859-018-2280-5

Zhang, Y. (2008). I-tasser server for protein 3D structure prediction. *BMC Bioinformatics* 9:40. doi: 10.1186/1471-2105-9-40

Zhou, J., and Troyanskaya, O. (2014). "Deep supervised and convolutional generative stochastic network for protein secondary structure prediction," in *International Conference on Machine Learning* (Beijing: PMLR), 745–753.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read
for greatest visibility
and readership

FAST PUBLICATION
Around 90 days
from submission
to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative,
and constructive
peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers
acknowledged by name
on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: frontiersin.org/about/contact

REPRODUCIBILITY OF RESEARCH
Support open data
and methods to enhance
research reproducibility

DIGITAL PUBLISHING
Articles designed
for optimal readership
across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics
track visibility across
digital media

EXTENSIVE PROMOTION
Marketing
and promotion
of impactful research

LOOP RESEARCH NETWORK
Our network
increases your
article's readership